



---

Supervision and control XML-based  
from Windows Vista to Windows CE

# Programmer Guide

---

Version 11 - Ed. Mar 2008

Cod. DOCS 11 DEV-E Build 1010

# Table Of Contents

<b>1. INTRODUCTION .....</b>	<b>13</b>
INSTALLING AND RUNNING APPLICATION .....	13
NEW FEATURES IN MOVICON 11 .....	17
MOVICON POWERHMI LIMITATIONS .....	22
MOVICON BA RESTRICTIONS (BUIDLING AUTOMATION) .....	25
<b>2. IMPORTANT INFORMATION .....</b>	<b>27</b>
ASSUMPTIONS.....	27
CONVENTIONS.....	27
COPYRIGHTS .....	28
WARNING.....	28
WINDOWS VISTA SUPPORT.....	29
TECHNICAL SUPPORT .....	30
<i>Web Support Center</i> .....	30
UPDATES AND BUGBASE .....	31
<i>Updates</i> .....	31
<i>Bugbase</i> .....	31
<b>3. BEFORE STARTING.....</b>	<b>33</b>
MICROSOFT WINDOWS CERTIFIED .....	38
PROJECT PROGRAMMING LIMITS.....	39
<b>4. PROJECTS STRUCTURE.....</b>	<b>43</b>
A NEW PROJECT .....	44
<i>Project Name</i> .....	44
<i>Users</i> .....	45
<i>Communication Drivers</i> .....	47
<i>Screens</i> .....	47
<i>DataBase Settings (ODBC)</i> .....	47
<i>Data Loggers and Recipe Settings (ODBC)</i> .....	49
<i>Alarms Settings</i> .....	50
PROJECT CONSTRAINTS .....	51
NEW PROJECT WIZARD CUSTOMIZING .....	54
THE WORKSPACE .....	55
<i>Docking Project Windows</i> .....	56
<i>The Rulers</i> .....	58
<i>The Screen Grid</i> .....	59
<i>The Movicon Grid</i> .....	59
<i>Customizing Menus and Toolbars</i> .....	60
<i>Resource TAB Groups</i> .....	61
TOOL BARS.....	62
<i>Tool Bar</i> .....	62
<i>Aligning Bar</i> .....	66
<i>Symbol Bar</i> .....	67
<i>Layer Bar</i> .....	67
<i>Formatting Bar</i> .....	68
<i>Status Bar</i> .....	69
<i>RunTime Bar</i> .....	70
MENUS .....	72
<i>File Menu</i> .....	72
<i>Edit Menu</i> .....	74
<i>View Menu</i> .....	75
<i>Insert Menu</i> .....	77
<i>Layout Menu</i> .....	77
<i>Symbols Menu</i> .....	78
<i>Tools Menu</i> .....	81
<i>Window Menu</i> .....	82

<i>Help (?) Menu</i> .....	83
<i>Context Resource Menus</i> .....	84
<i>Tools Menu Configuration</i> .....	87
PROJECT EXPLORER WINDOW .....	89
PROPERTIES WINDOW .....	90
LOGIC EXPLORER WINDOW .....	93
TOOLBOX WINDOW .....	96
SYMBOLS LIBRARY WINDOW .....	97
DYNAMIC PROPERTY INSPECTOR WINDOW .....	100
REFACTORING EXPLORER WINDOW .....	102
XML CODE EXPLORER WINDOW .....	103
TAG BROWSER WINDOW .....	104
LIST VARIABLES WINDOW .....	105
OUTPUT WINDOW .....	107
DYNAMIC HELP WINDOW .....	108
WATCH WINDOW .....	109
POPUP WINDOW .....	111
THE RESOURCES .....	112
<i>The Multiple Project Resources</i> .....	112
<i>Common General Resource Properties</i> .....	113
<i>Resource Folder General Properties</i> .....	113
PROJECT FILES .....	114
<i>Renaming Resources</i> .....	118
PROJECT SETTINGS .....	120
<i>General Project Settings</i> .....	121
<i>Project Platform Settings</i> .....	122
<i>Project Path Settings</i> .....	122
<i>Project Execution settings</i> .....	124
<i>Heap Manager Settings</i> .....	126
<i>Project Historical Log Settings</i> .....	129
<i>Child Project Options Settings</i> .....	131
<i>Project Spooler Manager Settings</i> .....	132
<i>tools</i> .....	133
VISUAL STUDIO SOURCESAFE 2005 INTEGRATION AND SUPPORT .....	136
<i>SourceSafe in Symbol Libraries</i> .....	138

## 5. REAL TIME DB..... 139

VARIABLE LIST (TAGS) .....	141
VARIABLE TRACING (AUDIT TRAIL) .....	142
VARIABLE SHARING WITH ODBC .....	143
COPY&PASTING VARIABLES .....	145
SYSTEM VARIABLES .....	145
VARIABLE AREAS .....	152
<i>Retentive Variables and Statistical Data</i> .....	152
<i>Renaming Variables</i> .....	154
REAL TIME DB SETTINGS .....	156
<i>Real Time DB General Settings</i> .....	156
<i>Real Time DB Retentive Data Settings</i> .....	157
<i>Real Time DB InUse Variable Manager Settings</i> .....	157
<i>Real Time DB OPC Server Settings</i> .....	158
<i>Real Time DB OPC Server Security Settings</i> .....	159
<i>Real Time DB OPC XML DA Server Settings</i> .....	159
<i>Real Time DB Trace DB Settings</i> .....	160
<i>ODBC Real Time I/O Link Settings</i> .....	162
DYNAMIC VARIABLES .....	163
<i>Dynamic Network Variables</i> .....	164
<i>Dynamic OPC Variables</i> .....	165
<i>Dynamic Communication Driver Variables</i> .....	166
STRUCTURE VARIABLES .....	167
<i>Structure Prototypes General Properties</i> .....	169
<i>Variable Member Order</i> .....	169
<i>Pointing to Structure Variables</i> .....	170
<i>Default Structures in symbols</i> .....	170
VARIABLE PROPERTIES .....	171

<i>General Variable Properties</i> .....	172
<i>Variable Engineering Data Properties</i> .....	174
<i>Variable Access Level Properties</i> .....	177
<i>Variable Options Properties</i> .....	178
<i>Variable Trace Options Properties</i> .....	179
<i>Variable ODBC Real Time I/O Link Properties</i> .....	181
<i>Variable Network Client Properties</i> .....	181
ASSOCIATING ALARMS TO VARIABLES .....	183
ASSOCIATING DATA LOGGERS AND RECIPES TO VARIABLES .....	183
ASSOCIATING EVENTS TO VARIABLES .....	183
GROUPING VARIABLES .....	184
POINTING TO A VARIABLE'S BIT .....	184
<b>6. SCREENS .....</b>	<b>185</b>
INSERTING SCREENS .....	186
STARTUP SCREEN .....	186
OPENING SCREEN MODALITIES .....	187
SCREENS WITH PARAMETERS .....	189
LOCAL SCREEN VARIABLES .....	191
NUMERIC AND ALPHANUMERIC SCREENS .....	192
SCREEN CHANGE COMMANDS BASED ON SCREEN IDS .....	193
SCREEN RESIZE .....	194
SCREEN NAVIGATION EDITOR .....	194
<i>Screen Navigation Style Properties</i> .....	196
SCREEN PROPERTIES .....	197
<i>Screen General Properties</i> .....	197
<i>Screen Style Properties</i> .....	198
<i>Screen Background Properties</i> .....	199
<i>Screen Execution Properties</i> .....	201
ASSOCIATING VARIABLE SCRIPT EVENTS TO SCREENS .....	201
<b>7. CROSS REFERENCE .....</b>	<b>203</b>
VARIABLE CROSS REFERENCE .....	203
SCREEN CROSS REFERENCE .....	204
<b>8. GRAPHIC INTERFACE .....</b>	<b>205</b>
HOW TO CREATE GRAPHIC INTERFACES .....	206
GRAPHIC EDITING OPERATIONS .....	210
<i>Tab Order</i> .....	210
<i>Overlap Order</i> .....	210
<i>Multi-Object Selections</i> .....	211
<i>Object Alignment Functions</i> .....	211
<i>Distribute Object Space</i> .....	213
<i>Find and Replace symbol</i> .....	214
<i>Importing/Exporting Vectorial drawings</i> .....	216
<i>Embedding Images in Symbols</i> .....	216
<i>Locking-Unlocking Objects</i> .....	217
<i>Colours Selection</i> .....	217
EDITING SYMBOL GRAPHICS .....	219
<i>Commands for Creating Symbols</i> .....	220
TEMPLATES: DYNAMIC SYMBOLS .....	221
CREATING OR EDITING SYMBOLS .....	221
<i>Inserting Symbols into the Library</i> .....	222
<i>Inserting Symbols on Screen</i> .....	223
<i>Script Code Editing</i> .....	223
<i>Variables inside Symbols</i> .....	224
<i>Images inside Symbols</i> .....	224
<i>Dropping Template Code</i> .....	224
<b>9. DRAWS AND OBJECTS .....</b>	<b>227</b>
<i>Tooltips in Controls</i> .....	227
BASIC SHAPES .....	228



OBJECT EDITING COMMAND SHORTCUTS .....	230
<i>Symbol Configuration Files</i> .....	233
<i>Object Default Settings File</i> .....	234
<i>Outside Screen Objects</i> .....	239
BUTTONS AND HOT REGIONS .....	240
<i>Components deriving from Button Objects</i> .....	241
<i>Button and Hot Region Execution Properties</i> .....	242
<i>Button Background Attributes Properties</i> .....	244
THE GAUGE.....	245
DUNDAS GAUGES .....	246
<i>Components deriving from the Gauge object</i> .....	247
<i>Gauge Variables Properties</i> .....	249
<i>Gauge Style Properties</i> .....	250
<i>Gauge Circular Setting Properties</i> .....	251
<i>Gauge Scale Settings Properties</i> .....	252
<i>Gauge Warning Zone Properties</i> .....	253
EMBEDDED SCREENS .....	254
<i>Embedded Screen Style Properties</i> .....	254
DISPLAYS .....	256
<i>Display Variables Properties</i> .....	256
<i>Display Style Properties</i> .....	258
THE LIST OBJECTS.....	261
<i>ListBox Execution Properties</i> .....	261
<i>List Style Properties</i> .....	262
COMBO BOX OBJECTS .....	262
<i>Combo Box Style Properties</i> .....	263
<i>Combo Box Variables Properties</i> .....	263
TAB GROUP OBJECT .....	264
<i>TAB Group Object Features and Constraints</i> .....	265
<i>TAB Group Style Properties</i> .....	265
<i>TAB Group Execution Properties</i> .....	266
ALARM WINDOWS.....	267
<i>Alarm Window Fields</i> .....	267
<i>Alarm Window Field Choice</i> .....	269
<i>Alarm Help in the Alarm Window</i> .....	270
<i>Alarm Window Style Properties</i> .....	270
<i>Alarm Window Execution Properties</i> .....	276
LOG WINDOW .....	277
<i>Log Window Fields</i> .....	277
<i>Log Window Field choice</i> .....	278
<i>DataBase Fields</i> .....	279
<i>Log Window Style Properties</i> .....	280
<i>Log Window Execution Properties</i> .....	282
<i>Log Window Data Filter</i> .....	283
THE TRENDS.....	285
DATA ANALYSIS .....	287
<i>Trend Functions</i> .....	289
<i>Linking Trends or Data Analysis to the Data Logger</i> .....	290
<i>Displaying Trend data</i> .....	291
<i>Extended Functionality with Basic Script Interface</i> .....	291
<i>Preconfigured Shortcut Keys</i> .....	291
<i>Mouse Executable Operations in RunTime</i> .....	292
<i>Change Language for Pens</i> .....	293
<i>Interactive Legend Area</i> .....	293
<i>Variable Quality Management</i> .....	294
<i>Trend and Data Analysis Settings</i> .....	294
<i>Trend and Data Analysis Style Properties</i> .....	294
<i>Trend and Data Analysis Pen Properties</i> .....	298
<i>Trends Pen Area Properties</i> .....	302
<i>Trend and Data Analysis Legend Area Properties</i> .....	302
<i>Trend and Data Analysis Time Area Properties</i> .....	303
<i>Trend and Data Analysis Grid Properties</i> .....	305
<i>Trend and Data Analysis Execution Properties</i> .....	306
<i>Trend Variables Properties</i> .....	308
<i>Templates with Trends and Data Analysis</i> .....	310
CHART OBJECTS.....	311

Chart Designer .....	311
Editing Data and Labels .....	315
Chart Wizard .....	315
Chart Properties .....	317
Chart General Properties .....	317
Chart Execution Properties .....	318
Chart Style Properties .....	319
DATALOGGER WINDOW .....	320
DataLogger Window Fields .....	320
DataLogger Window Field Choice .....	321
DataLogger Window Style Properties .....	322
DataLogger Window Execution Properties .....	323
DataLogger Data Filter .....	324
TRACEDB WINDOW .....	325
TraceDB Window Fields .....	325
TraceDB Window Field Choice .....	326
TraceDB Window Style Properties .....	327
TraceDB Window Execution Properties .....	328
TraceDB Data Filter .....	329
THE GRID .....	330
The Grid Buttons .....	330
Grid Style Properties .....	331
Grid Execution Properties .....	332
Data Source Selection (DSN) .....	333
HOURL SELECTOR .....	336
Hour Selector Style Properties .....	338
Hour Selector Execution Properties .....	341
IP CAMERA VIEWER .....	342
The IP Camera Viewer General Properties .....	343
IP Camera Execution Properties .....	343
ACTIVEX/OCX OBJECTS .....	344
OLE OBJECTS .....	344
DRAWINGS AND CONTROLS COMMON PROPERTIES .....	345
Variable Properties .....	345
Style Objects Properties .....	346
Execution Object Properties .....	348
General Object Properties .....	349
Visible Properties Animation .....	352
Transparency Animation Property .....	353
Composed Movement Animation .....	353
Move X Properties Animation .....	355
Move Y Properties Animation .....	356
Points Properties Animation .....	357
Start X Point Properties Animation .....	358
Start Y Point Properties Animation .....	359
End X Point Properties Animation .....	360
End Y Point Properties Animation .....	361
Scaling Properties Animation .....	362
Rotation Properties Animation .....	363
Text Properties Animation .....	364
Back Color Properties Animation .....	366
Edge Color - Dynamic Text Animation .....	367
Filling Properties Animation .....	367
Colour Threshold Settings .....	369
Images Properties Animation .....	371
Visibility Properties .....	373
Position Properties .....	374
Stroke Attributes Properties .....	375
Background Attributes Properties .....	375
Access Levels Properties .....	376
Font Properties .....	377
Script Debug Output Properties .....	378
Dragging Properties .....	378
ASSOCIATING VARIABLE SCRIPT EVENTS TO SYMBOLS .....	380
PUBLIC SYMBOLS .....	381
STYLE SOURCES IN SYMBOLS .....	382

<b>10. ACTIVEX OBJECT .....</b>	<b>387</b>
ACTIVE X OBJECTS .....	387
INSERTING AN ACTIVEX .....	388
ACTIVE X EDITING.....	389
EDITING ACTIVEX USING VBA .....	389
ACTIVE X PROPERTIES .....	391
<i>Active X Execution Properties.....</i>	<i>391</i>
<i>ActiveX General Properties.....</i>	<i>392</i>
OLE OBJECTS .....	392
<i>Inserting OLE objects .....</i>	<i>392</i>
<i>Editing OLE objects .....</i>	<i>393</i>
<b>11. MENU.....</b>	<b>395</b>
INSERTING MENUS.....	396
MENU ITEMS.....	396
<i>Test Menus.....</i>	<i>398</i>
CUSTOMIZING MENU BARS .....	398
MENU PROPERTIES.....	398
<i>Menu General Properties.....</i>	<i>398</i>
<i>Item Menu General Properties .....</i>	<i>399</i>
<i>Menu Item Execution Properties.....</i>	<i>399</i>
MENU EXAMPLE .....	400
<b>12. SHORTCUTS .....</b>	<b>401</b>
SHORTCUT ACTIVATION .....	402
INSERTING SHORTCUTS .....	402
SHORTCUT KEYBOARD COMMANDS .....	403
SPEECH RECOGNITION.....	404
SHORTCUT PROPERTIES .....	406
<i>Shortcut General Properties.....</i>	<i>406</i>
<i>Shortcut Command General Properties.....</i>	<i>406</i>
<i>Shortcut Command Execution Properties.....</i>	<i>407</i>
SHORTCUT EXAMPLE .....	409
<b>13. STRING TABLE .....</b>	<b>411</b>
INSERTING STRINGS.....	411
<i>Inserting Missing Strings.....</i>	<i>412</i>
STRING SELECTION WINDOW.....	413
CHANGING LANGUAGES .....	413
STRING FORMATTING THROUGH STRING IDS.....	415
<i>Change System Language.....</i>	<i>415</i>
LANGUAGE COLUMN PROPERTIES .....	418
<i>General Language Column Properties.....</i>	<i>418</i>
<b>14. VARIABLES SCALING .....</b>	<b>419</b>
VARIABLE SCALING PROPERTIES.....	420
<i>Variable Scaling General Properties .....</i>	<i>420</i>
NON LINEAR SCALING .....	422
<b>15. COMMAND SCHEDULER .....</b>	<b>423</b>
<i>Holiday Planning.....</i>	<i>424</i>
COMMAND SCHEDULER PROPRIETIES.....	426
<i>Command Scheduler General Properties .....</i>	<i>426</i>
<i>Scheduler Access Level Properties.....</i>	<i>428</i>
COMMAND SCHEDULER EXAMPLE .....	429
<b>16. COMMANDS ON EVENT.....</b>	<b>431</b>
EVENT OBJECT PROPERTIES.....	431
<i>Event Objects General Properties .....</i>	<i>432</i>

EVENTS AS TEMPLATES .....	433
<b>17. OBJECTS COMMANDS LIST .....</b>	<b>435</b>
<i>Variable Commands</i> .....	435
<i>Screen Commands</i> .....	439
<i>Script Command</i> .....	442
<i>User Commands</i> .....	443
<i>Report-Recipe Commands</i> .....	444
<i>Menu Commands</i> .....	446
<i>System Commands</i> .....	447
<i>Language Commands</i> .....	448
<i>Help Command</i> .....	448
<i>Event Commands</i> .....	448
<i>Alarm Commands</i> .....	449
<b>18. ALARMS .....</b>	<b>451</b>
INSERTING ALARMS .....	453
ALARMS HISTORICAL .....	455
ALARMS AS TEMPLATES .....	456
ALARMS ACTIVATED BY HOMONYMOUS VARIABLES .....	457
ALARM STATUS FILE .....	458
ALARM PROPERTIES .....	458
<i>Alarm General Properties</i> .....	458
<i>Alarm Threshold General Properties</i> .....	460
<i>Alarm Threshold Execution Properties</i> .....	462
<i>Alarm Threshold Style Properties</i> .....	463
<i>Alarm Threshold Event Notification Properties</i> .....	465
ALARM DISPATCHER .....	469
<b>19. HISTORICAL LOG .....</b>	<b>471</b>
LOG FILE MANAGEMENT .....	472
<i>Database Log File</i> .....	473
<b>20. HISTORICAL LOG MANAGEMENT .....</b>	<b>475</b>
ODBC STANDARD .....	476
<i>ODBC Drivers</i> .....	477
<i>ODBC Links</i> .....	478
<i>Connection Pooling</i> .....	480
<i>Creating ODBC links automatically</i> .....	480
<i>Creating ODBC Links Manually</i> .....	480
<i>SQL Server with MSDE</i> .....	482
<i>Using the ORACLE</i> .....	484
IMDB (IN MEMORY DB) .....	485
<i>IMDB Settings for Recording data</i> .....	486
<i>IMDB Database Limitations</i> .....	486
<i>DBMS Settings for Recording data on DataBase</i> .....	488
<b>21. USERS AND PASSWORDS MANAGEMENT .....</b>	<b>489</b>
USER LEVELS AND ACCESS LEVELS .....	490
CUSTOM PASSWORD DIALOG WINDOWS .....	492
GENERAL PROPERTIES COMMON TO USERS AND USER GROUPS .....	493
USERS AND USER GROUPS PROPERTIES .....	495
<i>Users and User Groups General Properties</i> .....	495
<i>Users and User Groups Windows User Properties</i> .....	496
<i>Users and User Groups Child Project Option Properties</i> .....	497
USER GROUP PROPERTIES .....	498
<i>User Group General Properties</i> .....	498
USERS PROPERTIES .....	499
<i>Users General Properties</i> .....	499
<i>User Data Properties</i> .....	500
RUNTIME USERS MANAGER .....	501
CFR21 PART 11 GENERAL CONCEPTS .....	503

VALIDATION CONFIGURING TECHNIQUES .....	505
<b>22. DATA LOGGERS AND RECIPES.....</b>	<b>511</b>
<i>The DataBase .....</i>	<i>512</i>
<i>DataBase Structure.....</i>	<i>513</i>
<i>Database Link to Report Files.....</i>	<i>513</i>
<i>Displaying Database Data.....</i>	<i>514</i>
DATA LOGGERS AS TEMPLATES .....	514
DATA LOGGERS AND RECIPES EDITOR .....	516
INSERTING DATA LOGGERS AND RECIPES .....	516
<i>Data Loggers and Recipe ODBC Link .....</i>	<i>517</i>
<i>Importing Data Loggers and Recipes from other Recipes.....</i>	<i>517</i>
RECIPES ON DATABASE.....	518
<i>SQL Commands for Recipes in the Database .....</i>	<i>519</i>
<i>Creating Recipes .....</i>	<i>520</i>
<i>The Query in Recipes.....</i>	<i>522</i>
DATA LOGGERS AND RECIPES COMMON PROPERTIES .....	522
<i>Data Loggers and Recipes common General Properties .....</i>	<i>523</i>
<i>Data Loggers and Recipes common Execution Properties .....</i>	<i>523</i>
<i>Data Loggers and Recipes common Database Settings Properties.....</i>	<i>524</i>
<i>Data Loggers and Recipes common Column General Properties.....</i>	<i>525</i>
DATA LOGGER PROPERTIES .....	526
<i>Data Logger Style Properties.....</i>	<i>526</i>
<i>Data Logger Recording Time Properties.....</i>	<i>528</i>
<i>Data Logger Execution Properties.....</i>	<i>529</i>
<i>Data Logger Column General Properties .....</i>	<i>529</i>
RECIPE PROPERTIES .....	531
<i>Recipe Execution Properties .....</i>	<i>531</i>
<i>Recipe Database Settings Properties .....</i>	<i>533</i>
<i>Recipe column General Properties .....</i>	<i>534</i>
<b>23. REPORTS .....</b>	<b>535</b>
REPORT DESIGNER.....	535
<i>How to create a Report.....</i>	<i>537</i>
CRYSTAL REPORT™ .....	545
CREATING REPORTS WITH CRYSTAL REPORT .....	545
<b>24. IL LOGIC (INSTRUCTIONS LIST) .....</b>	<b>549</b>
INSERTING IL LOGIC INSTRUCTIONS .....	549
LOGIC PROGRAMMING .....	551
<i>Combined Logic Result CLR .....</i>	<i>551</i>
<i>Bracket levels for the CLR.....</i>	<i>552</i>
<i>Jump to Labels.....</i>	<i>552</i>
<i>Variables in Floating Point.....</i>	<i>553</i>
<i>The Logic Accumulators.....</i>	<i>553</i>
<i>Constants .....</i>	<i>554</i>
<i>End program processing.....</i>	<i>554</i>
<i>Inserting Comments.....</i>	<i>554</i>
LOGIC BIT INSTRUCTIONS .....	555
<i>AND Logic.....</i>	<i>555</i>
<i>OR Logic.....</i>	<i>555</i>
<i>Sequence Combination.....</i>	<i>555</i>
<i>Setting Variables .....</i>	<i>555</i>
<i>Setting Timers .....</i>	<i>556</i>
<i>Summary Table Bit Instructions .....</i>	<i>557</i>
WORD LOGIC INSTRUCTIONS .....	559
<i>The ACCU1 and ACCU2 Logic Accumulators.....</i>	<i>559</i>
<i>Loading Variables and Constants.....</i>	<i>559</i>
<i>Transferring Variables.....</i>	<i>559</i>
<i>Logic Operations on Word Variables .....</i>	<i>559</i>
<i>Operations on Accumulators.....</i>	<i>560</i>
<i>Word Summary Table .....</i>	<i>560</i>
COMPARE INSTRUCTIONS.....	563

<i>Compare Summary Table</i> .....	563
ARITHMETIC INSTRUCTIONS .....	565
<i>Arithmetic Instructions Summary Table</i> .....	565
MATHEMATIC INSTRUCTIONS .....	568
<i>Mathematic Instructions Summary Table</i> .....	568
<b>25. SYNAPSES .....</b>	<b>573</b>
<i>Inserting Synapse</i> .....	574
<i>Connecting Synapses</i> .....	575
<i>Synapse Logic Programming</i> .....	577
SYNAPSE EXECUTION .....	578
<b>26. SOFT LOGIC IEC 1131 .....</b>	<b>581</b>
<i>Soft Logic General Properties</i> .....	582
<i>Variable Database Synchronization</i> .....	583
<i>ProConOs OPC Server Configuration</i> .....	584
<b>27. CHILD PROJECTS .....</b>	<b>587</b>
<i>Child Project Paths</i> .....	590
<i>Child Project Resource Access</i> .....	590
<i>Strings in Child Projects</i> .....	592
<i>Child Project Users Inclusion</i> .....	592
<b>28. NETWORKING .....</b>	<b>593</b>
<i>Network Installation</i> .....	594
<i>Network Connection</i> .....	595
<i>Windows Terminal Server</i> .....	595
<i>Remote Project Debugging</i> .....	596
NETWORK SERVICES PROPERTIES .....	597
<i>Network Server Properties</i> .....	597
<i>Log and Security Properties</i> .....	598
<i>Local Transport Properties</i> .....	599
<i>TCP Transport Properties</i> .....	599
<i>TCP Transport Advanced Settings Properties</i> .....	600
<i>TCP Transport SOCKS Options Properties</i> .....	601
<i>UDP Transport Properties</i> .....	601
<i>UDP Transport Advanced Settings Properties</i> .....	602
<i>HTTP Transport property</i> .....	602
<i>HTTP Transport Advanced Settings Properties</i> .....	603
<i>HTTP Transport SOCKS Options Properties</i> .....	604
SETTING UP NETWORK CLIENTS .....	605
<i>Network Client Properties</i> .....	605
<i>Network Client General Properties</i> .....	606
<i>OPC XML DA Client Properties</i> .....	606
RAS STATIONS .....	607
<i>RAS Stations General Properties</i> .....	607
CLIENT RULES .....	608
<i>Client Rules General Properties</i> .....	609
<b>29. REDUNDANCY .....</b>	<b>611</b>
<i>Redundancy Levels</i> .....	612
<i>Redundancy Functionality</i> .....	612
<i>Advanced Redundancy Functions</i> .....	614
REDUNDANCY SETTINGS .....	616
<b>30. OPC (OLE FOR PROCESS CONTROL) .....</b>	<b>619</b>
OPC CLIENT EDITOR .....	619
<i>OPC Client DA Properties</i> .....	620
<i>OPC Server DA Properties</i> .....	621
<i>Entering New OPC Servers</i> .....	621
<i>OPC Item Connection String</i> .....	622
<i>Entering New Groups</i> .....	623

Entering New Items .....	624
OPC SERVER.....	625

### **31. ON-LINE CHANGES TO PROJECTS .....627**

List of editable resources in RunTime.....	627
--	-----

### **32. MOVICON CONFIGURATION KEYS.....629**

CONFIGURATION KEY LIST .....	631
REGISTRY KEYS DEVELOPMENT REGISTRY KEYS .....	631
USER INTERFACE REGISTRY KEYS USER INTERFACE .....	636
REGISTRY KEY COMMUNICATIONS.....	640
REGISTRY KEYS HISTORICALS .....	642
GENERAL REGISTRY KEYS GENERALS .....	645
ADVANCED REGISTRY ADVANCED KEYS.....	647
SPEECH SYNTHESIS KEYS .....	649
SOURCESAFE KEYS .....	649
MS ACTIVE SYNC KEYS .....	652

# 1. Introduction

---

## ***Movicon Programmer's Guide Contents***

The Programmer's Guide contains all the necessary information for Movicon developers. The chapters in this guide describe the projects, the Resources, the Objects and the techniques for logical and graphic programming to permit the optimal use of the system's resources in developing projects.

The manual describes how to program and structure projects, how to use the logic editor, how to animate screens, how to do data or recipes settings by building dialog boxes, how to use objects and any other item which constitutes a system resource.

One chapter has been dedicated on how to use the Movicon vectorial graphic editor and the default Bitmap editor, how to use the standard symbol library which Movicon places at the user's disposal, how to import symbols or strings using the "Drag & Drop" techniques.

The Movicon supervision package fully complies with the Windows standard, therefore the reader is advised to consult the Ms Windows™ 'User's Manual' for further information on windows, menus and dialog boxes.



Any corrections or updates that may have been made to this manual can be referred to in the "Readme.txt" file contained in the installation CD-ROM and installed in the Movicon user directory.

## Installing and Running application

---

The procedures for installing, running up and configuring Movicon for correct system usage are described below.

The Microsoft Windows installation already provides a operating system configuration for adapting to the hardware platform. However, it would be to your benefit to follow the few indications below for further optimization.

### **Installing Movicon**

---

The procedures for installing Movicon on the hardware platform are very simple, and comply with the Microsoft Windows specifications. When inserting the CD-ROM you will automatically enter into the installation environment where the guide command will activate for installing the Movicon components desired. When you wish to install Movicon manually from the CD, you can access the CD's folder from the Windows Explorer and execute the "Setup.exe" file contained in it, for example:

D:\Movicon\setup\Setup.exe

The Setup will carry out the Movicon installation on the PC, creating a group of Movicon icons upon terminating which can be accessed with the "Start" button from the Windows application bar. The Movicon Group, in icon form, will contain the shortcut to the Movicon .exe file. During installation, which takes a few minutes, you will be asked to enter the name of the folder in which files necessary to Movicon are to be organized. When you do not wish to change the name or path, simply confirm the "Movicon11" default folder which will then be created with the path: "..\Program Files\Progea\Movicon11".

In the first installation phases you will be asked for the installation type you wish to executed.

The options are:



- **Typical:** installs the most commonly used components without requiring anything from the user (recommended)
- **Custom:** consents the user to select the components to install (recommended for expert users only)

## Uninstalling Movicon

---

Movicon, in compliance with the Windows standards, consents to a simple system uninstalling to remove all its files from the PC.

To proceed with uninstalling Movicon simply activate the appropriate icon from the Windows Control Panel for installing/removing applications.

The uninstalling procedure will completely remove all the Movicon files upon receiving confirmation.

## Restore Default settings

---

All default settings will be restored if you press the "CTRL" key while starting up Movicon and keep it pressed down until Movicon has completely started up. This will restore the work area availability (windows, menus etc.) with their default parameters. This command must also be used when needing to change languages in the Movicon development environment using the "LangManager" tool.

## Movicon Execution

---

Movicon is a software platform which allows you to plane supervision applications and run them. Movicon therefore works in two modes:

- Project Designing (Development)
- Running Projects (Runtime)

The platform is installed as a executable application available in two versions:: Movicon.exe and MoviconRunTime.exe. Movicon.exe consents both project development and runtime execution, while MoviconRunTime.exe consents runtime execution only.



If not specified otherwise, the information given here refers to Movicon.exe, the product version which consents both development and runtime execution. Only the runtime execution engine (MoviconRunTime.exe). can be installed in plant systems.

## Movicon Startup

---

The Movicon installation will create a group of icons which can be accessed from the Windows' Start menu. When running the Movicon .exe without specifying any options will enter you into the Movicon environment in programming mode, with the automatic opening of a new project or with he last opened project.

By using the appropriate syntaxes in the Options for the command line at the Movicon startup you will be able to change the system's default settings, associate project files, custom modules and other options as explained below. The syntax for starting up Movicon with the command options is:

**Movicon /[Options] <project file>**

If you wish to startup Movicon and run a previously programmed project at the same time, you need to use the /R. option. Below an example is given for automatically running the TEST.MOVPRJ project file:

**C:\Program Files\Progea\Movicon11\Movicon.exe /R  
C:\Documents\Test.movprj**

If you wish to add or change the command options or the Movicon working folder in Windows, use the following procedures:

Press the right mouse key in the workspace of the Windows Desktop and select the "New - Shortcut" command. Then write the desired command line for the Movicon Startup in the settings window which appears.

## Startup Movicon in RunTime Mode

Once you have created the project it can be run directly without having to pass through the development environment. In order for this to happen you must create a link/connection, as explained above, to the Movicon Runtime Module being the "MoviconRunTime.exe" file which is found in the same Movicon installation folder. The following example is a command line for automatically running the TEST.MOVPRJ project file:

```
C:\Program Files\Progea\Movicon11\MoviconRunTime.exe  
C:\Documents\Test.movprj
```

You can get the same result by using the "Movicon.exe." followed by the "/R" option as already explained above.

## Movicon Startup as Service

The Windows Services are applications which are run automatically at the Windows startup, before and independently from User log on. Movicon fully supports the Windows Services as described below:

When Movicon is started up as Service, after the operating system LogOn, the user interface will not be displayed but will be shown as a Movicon icon in the Windows application bar. To open the project's user interface double click this icon.



In order to make the project's user interface open automatically after Windows user login you will have to enable the project's "Show Service at Log On" execution property.

You can install Movicon as Service in addition to the development environment by using the "Install this Project as Windows Service" command and also by means of the Windows command line using the "/i" or "-i" option. By executing the command line from the Windows "Start - Run" menu:

**MoviconService.exe -i**

Movicon will be installed as Service. You must then add the parameter, which identifies the project that the service must run, in the Windows configuration Registry. The key to be modified is:

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MOVICON  
X\Parameters]  
"Project"="C:\\Project Path\\ProjectName.movprj"
```

The text above can be copied in a text file with a ".reg." extension in order to automatically create the key by double-clicking on the ".reg" file to insert key in the Windows registry. To insert the key manually you will have to go to the following parameter group:

```
"[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MOVICON X\Parameters]"
```

and create the "Project" key string type inserting the "C:\\Project Path\\ProjectName.movprj" value.

## **"Forced" Startup in Demo Mode**

---

Running the application with the "D" key pressed, Movicon will start up in Demo Mode. This function is used for modifying certain places in the project which have a Runtime software or hardware license only in order not to replace this licence. This function has been created for the sole purpose of modifying projects by remote control (with PCAnywhere type applications) where it is impossible to remove the license from the PC if too far away.

In addition, you can save projects when the inserted license, hardware or software, is runtime only with Movicon already started up normally. In this case you will need to keep the "D" key pressed down and execute the project save command. This will enter the application into "Demo Mode" saving the project. To escape from the "Demo Mode" you will have to close and reopen Movicon.

## **Using Windows Terminal Server**

---

Movicon supports the Windows Terminal Server (or Remote Desktop) and consents more than one program instances to be run on the server. Using Movicon with Windows Terminal Server requires a "NET" license. Normal licenses will be ignored by Movicon if run from the Windows Terminal Serve consequently running program in Demo mode. For further information please refer to "Windows Terminal Server".

# New features in Movicon 11

---

Movicon has introduced some new and important features with its 11th version. The main ones are listed and briefly described below:

New Feature	Description
Refurbished Graphics	<p>the Workspace has been improved. You can now opt for a application 'look' in line with Office or Windows and memorize window statuses according to the resource opened. New windows have also been added such as the "Refactoring Explorer" and the "XML code Explorer".</p> <p>The "Project Explorer" window no longer shows the "Screens", "Menu", "Accelerator" "Basic Script" and "Parameter File" resources grouped together but separately in different categories.</p>
Refactoring Explorer Window	This window displays errors relating to the selected resource. These errors only refer to variables inserted in the properties of the resource, which not have been defined in the RealTimeDB or because object has gone outside the screen's boundary.
XML Code Explorer Window	This window displays xml code relating to the selected resource or object.
Cross Reference	<p>The cross reference has been completely revised. A window will now open in the workspace listing the areas in which each variable is used. If a variable has been used in an object containing a screen, when double-clicking on the link this screen will open with focus on data of the object in question.</p> <p>The Cross Reference is also available for screens and lets you know which object or resource opens the screen.</p>
List Variables (Tags)	This window displays variables in table form through the Project Explorer window consenting much quicker variable setting and search operations. You can also execute filters and groups through this window.
Variable/Resource Renaming Manager	This new function allows the automatic updating of variable or resource references in a project when the variable or resource is renamed.
Style Source Container for Symbols	The Style Source Container management for Symbols consents the activation of certain graphical styles for objects in screens based on a Style Source Container. In this way the object can inherit a different style, defined by the programmer in the Style Source Container, during Runtime mode independently from the style associated to the object in design mode. This style source can be activated by the user logged on, from an active language or the style source container screen. The properties that can be inherited are:

	<ul style="list-style-type: none"> <li>• back color</li> <li>• text-edge color</li> <li>• font</li> </ul>
VBA Expression Editor	This window helps the programmer enter basic script expressions in object properties. This editor is every handy because it provides those operations needed for entering expressions consenting a major control of any errors made.
Object Property Settings	A Toggle icon has been placed on the top bar on the right of the properties window allowing users to access the object's advanced properties from easy mode within a click. "Advanced" sub-groups have also been added to the properties window, where the most advanced settings can be entered and which are normally modified by the programmer.
String Table	<p>Additional properties have been inserted to each String Table column which consent you to assign each language with a TTS engine for vocal texts. A command has been added for automatically inserting missing texts.</p> <p>The window used for entering strings in objects has been modified to allow you to insert new strings as well.</p> <p>A command to activate languages has been added to the Movicon status bar.</p> <p>You can now link together more than one ID string in one text field.</p>
Screen Editing	<p>Additional information on the object's position and size is now available in the status bar.</p> <p>Selecting object components used to build symbols has been simplified.</p> <p>A tab has been added to the Dynamic Property Explorer window listing the variables used by symbols.</p> <p>The editor can now be opened with a command to modify images associated to screens.</p> <p>New Zoom mode.</p> <p>Objects stay visible even when outside the screen's boundary.</p>
Shared Programming (Visual Studio SourceSafe 2005 integration)	<p>The Visual Studio SourceSafe consents project development to be shared and managed by different users working together.</p> <p>The SourceSafe support is also available for sharing the "Symbol Library" categories.</p>
Uploading/Downloading Projects	<p>The uploading/downloading projects to and from WinCE service is now available for Win32, meaning that project can now be uploading to Win32 client stations.</p> <p>Projects can be down/uploaded from and to WinCE/Win32 devices.</p> <p>You will also find a new command in the project upload/download window for cancelling project on the device and you can also browse upload device paths.</p>

Project Wizards	Two new Wizards have been added to the window used for creating new projects for creating an "Empty Project" and a "Template Project". These Wizards are very handy when customizing.
Variable Statistics	A greater quantity of data statistics is now saved in the variable retentivity files. These data statistics can also be visualized 'Display' objects using the new "Value to Display" property.
Trend/Data Analysis Objects	<p>The Trend object has been enhanced. The legend is interactive for selecting pens better. The Date/Time format can be customized. The data displayed can take variable qualities into consideration.</p> <p>A new object has been added, Data Analysis, deriving from the Trend. Being a powerful graphical data analyzer function, the Data Analysis allows you to carry out analysis exclusively on historical data. You can carry out data comparisons, measures between two points and select custom time ranges. The Data Analysis is not limited in the number of records it can display (The Trend has a maximum limit of 10000 displayed records).</p>
Scheduler Window Objects	<p>New programming mode in table form.</p> <p>Possibility to add new schedulers in runtime.</p> <p>Possibility to change reference variables for the first scheduler "variable" command type.</p> <p>Possibility to change the setting value for the reference variable for the first scheduler "variable" command type.</p> <p>User access level management for the command listed above.</p>
IP Camera Viewer Object	<p>New IP Camera Viewer object. This object is used for viewing images taken by IP cameras. It allows you to visualize a sequence of images in realtime from a IP camera.</p> <p>Supports image downloading using http.</p> <p>Supports user authentication logon.</p> <p>Supports dynamic 'URLs.</p>
Dundas Gauge Object	New Dundas Gauges object added. These gauges are inserted using a wizard giving you the possibility to use more attractive and realistic gauges. These objects can be Circular, Linear or Numeric Indicators.
Button Objects	<p>New 'square' buttons have been added in 3D to look like Windows Vista button looks.</p> <p>You can now custom the images associated to these buttons base on three statuses which they can obtain (pressed, pressed down, released).</p>
Pipe Object	The new Pipe object is a polyline derivative. It can be custom filling coloured but it does not support "Gradual Filling".
Display Object	<p>Added option to display the numeric or alphanumeric pad when you click on editable displays.</p> <p>Added option to display historical data of linked</p>

	variables.
Custom Toolbox	The "Toolbox" can be extended with new custom symbol categories. This means that new categories can be added containing symbols created by the programmer.
Default Object Setting Files	Movicon consents those objects in the "Toolbox" to be customized to allow the programmer to modify the characteristics of simple objects without needing to create a completely new symbol and adding to the Symbol Library. In order to do this, just create a file with the name of "ObjectName.defxml" to contain the default settings of the object in question.
Symbol configuration Files	Each composed control or symbol, that can be inserted on screen, can be associated with a Configuration File to save their settings so that they can then be restored in design mode or Runtime mode when needed. This file will be saved in XML format with the ".xml" extension.
Object Transparency Properties	A new property has been added for screen objects to set their transparency. the transparency level starts from "0" to "255", where "0" makes the object completely transparent.
Object color thresholds	The color thresholds in the dynamic properties of screen objects can now be copied and pasted in different points using the two new "Copy" and "Paste" buttons.
Report Designer	The Report Designer is an extremely powerful tool integrated in Movicon to give users the possibility to independently create reports relating to the Movicon databases, without need to purchase any supplementary software packets. This tool allows users to perform Filter, Group, Data Summaries (totals, averages, etc.), Sub Reports. It uses .NET components so that Reports can be executed over the WEB.
SoftLogic IEC1131	Thanks to the partnership with the German KW-Software company, a major company in the SoftPLC sector, the Multiprog is now fully integrated in the Movicon platform, making the Progea product a complete and all-in-one environment, both for supervisory and Realtime I/O control.
Screen Navigation	By using the "Screen Navigation Editor" you can graphically define screen navigations. This means that you can create connections between different screens so that in runtime mode they automatically convert into a navigation bar displayed in different screens.
New Basic Script Engine	The Basic Script engine is now compatible with the "VB.NET" language and supports Unicode strings.

Speech Recognition	The new Speech Recognitions function is now available and consents commands to be activated vocally using speech texts associated to the Accelerator resource. Supports SAPI 5.1.
Change Language	You can now associate a default language to each user so that when they log on their assigned default language will automatically activate.
Alarms and Messages	A new Message Management. Now when you configure Alarms as Messages they will be indicated with a different icon and the events will be identified with the "MESSAGE ON" and "MESSAGE OFF" in the historical Log. The new Alarm log management consents rapid displaying of alarm transactions through the Alarms window. Alarm transactions displayed in groups in the Historical Log Window. New Filter for Priorities in Alarm Window. Additional Alarm management based on the total variable durations taken from variable statistics. when an alarm is not associated to any activation variable, Movicon will control is a variable exists in the RealTime DB with the same name when in runtime mode. If this variable exists, it will be used as the alarm's activation variable.
Alarm Statistics	The alarm statistic Reports have been modified and improved. The four available reports show the following data: <ul style="list-style-type: none"> <li>• Analysis based on Frequencies</li> <li>• Analysis based on durations</li> <li>• List sorted by date</li> <li>• List sorted by duration</li> </ul>
Movicon CE	Movicon CE fully supports SQL Server CE 3.0 (SQL Server Mobile) and therefore the following new features: <ul style="list-style-type: none"> <li>• Possibility to open a ".sdf" file directly with SQL Server 2005</li> <li>• Possibility of multi-accessing the same database at the same time</li> <li>• Automatic Database Compacting (Auto-Shrink)</li> <li>• Maximum size possible is 4GB</li> </ul> Fully supports images.
Web Client	You can now define maximum sizes with which screens are to be displayed by the Web Client.



# Movicon PowerHMI limitations

The PowerHMI version has any limitations in respect the complete version of Movicon. Below is reported the list of these limitations.

## The PowerHMI doesn't support:

- ODBC
- ActiveX
- OLE Objects
- ROT
- Reports Engine (Crystal Report)
- AlarmDispatcher
- Web Client
- Power Template containing basic script code



**The product documentation is the same for all versions Movicon and PowerHMI. Possible references in the help to functionality that, like described above, are not supported, must not be considered valid.**

Below is reported a more detailed comparative table:

Function	PowerHMI	Movicon
<i>RealTime DB</i>	Max. 2048 bytes	to unlimited
Scaling	Yes	Yes
ODBC Realtime	No	Yes
Trace DB	No	Yes
Data Structure	Yes	Yes
OPC Client DA	Yes	Yes
OPC Server DA	No	Yes
OPC Client XML DA	Yes	Yes
OPC Server XML DA	No	Yes
Networking	Yes	Yes
IntelliSense in Basic Script	No	Yes
<i>Graphic Interface</i>		
Vectorial Graphic Editor	Yes	Yes
Support for BMP, GIF, JPG, WMF, EMF	Yes	Yes
Dynamic Animation	Yes	Yes
Symbols Library	Yes	Yes
Import/Export Symbols	Yes	Yes
Public Symbols	No	Yes
Power Templates (VBA Symbols)	No	Yes
Charts 3D	No	Yes
Grid	No	Yes
Sinapsis	No	Yes
Scheduler	Yes	Yes
Editing Menu	Yes	Yes
Style Source Management in Symbols	No	Yes
<i>Alarms Logs</i>	Max. 1024 Alarms	unlimited
Alarms Management	Yes	Yes
Historical Management (XML)	Yes	Yes
Historical Management (ODBC)	No	Yes

Alarms Notification (SMS, Email, Voice)	No	Yes
Server OPC AE	No	Yes
Alarms Area	Yes	Yes
Statistic Reports	No	Yes
<i>Recipes - Data Loggers</i>		
Recipes / Data Logger (XML)	Max nr. 2	Yes
Recipes / DataLoggers (ODBC)	No	Yes
Report Engine	No	Yes
Text-To-Speech	No	Yes
<i>Trends</i>		
Trend RealTime	Yes	Yes
Historical Trends on file .CSV	Yes	Yes
Historical Trends (linked to DataLogger XML)	Yes	Yes
Historical Trends Database (ODBC)	No	Yes
<i>Users &amp; Passwords</i>		
Use 1024 levels	Yes	Yes
User Groups	Yes	Yes
CFR21	No	Yes
Runtime Users	Yes	Yes
<i>Dynamic Multilanguage</i>	Yes	Yes
Unicode Support	Yes	Yes
<i>Drivers</i>		
Drivers Library	Reduced	Full
Max. number of Driver	Max nr. 2	unlimited
PLC Tag Importer	Yes	Yes
<i>Event Object</i>	Yes	Yes
<i>Scaling Object</i>	No	Yes
<i>Scheduler Object</i>	Yes	Yes
<i>Logics</i>		
IL Logic (Step5-Step7)	Yes	Yes
VBA Logic (WinWrap Basic)	Reduced	Yes
Sinapsis Logic	No	Yes
<i>Networking</i>	Yes	Yes
<i>Redundancy</i>	No	Yes
<i>Child Projects</i>	No	Yes
<i>Screen Navigation</i>	No	Yes
<i>Visual Studio SourceSafe 2005 Integration</i>	No (the menu items will be disabled)	Yes
<i>Speech Recognition</i>	Yes	Yes
<i>Web Client</i>	No	Yes

<i>Services Support</i>	No	Yes
<i>Touch Screen Support</i>	Yes	Yes
<i>Cross Reference</i>	Yes	Yes
<i>Debugger</i>	Yes	Yes

## **VBA (WinWrap Basic) Restrictions**

---

The usage of the Basic Script resources has some restrictions:

- Only two Basic Script resources can be loaded in Ram and executed at the same time
- It's not possible to associate the script code to the Objects, Screens and Alarms Threshold
- The basic script expressions in the objects properties are not supported
- It's possible to insert the variable script events in the Basic Script
- The usage of Public Basic Scripts is supported
- The direct access to the project variables is supported
- The Basic Script Resources can not be run in "Safe Mode"
- The Basic Script Resources can access and use external libraries, but only libraries that don't need any licence
- All the WinWrap Basic functions are supported in the Basic Script Resources
- The IntelliSense for the project RealTimeDB variables is not supported. This also implies that it will not be possible to read/write structure variables using the <Variable>.<Member> syntax or a bytes array variable type with the <Variable>.<e>n syntax
- Only some functions of the "DBVariableCmdTarget" and "ScriptMEInterface" are supported in the Basic Script Resources. Find below the supported functions:

The supported methods and properties of the "DBVariableCmdTarget" interface are:

- EnableInUseVarMng
- GetDynamicVariable
- GetNumDynTag
- GetNumStructDefinitions
- GetNumVariables
- GetVariableAddressInfo
- GetVariableDescription
- GetVariableSize
- GetVariableValue
- GetXMLSettings
- IsFirstDBInstance
- IsValidVariable
- PurgeDynTag
- PurgeDynTagTimer
- SetVariableValue
- UseSharedDynTag
- VariableInUse

All the events, methods and properties of the "ScriptMEInterface" interface are supported.

## **Restrictions on System Variables**

---

The following system variables are not managed by PowerHMI:

\_SysVar\_:NumWebClientConnected  
\_SysVar\_:PendingTraceEntries

\_SysVar\_: NumExpressions  
\_SysVar\_: OPCServerNumClients

The command for adding the system variable to the database will not add these variable in the "\_SysVar\_" structure prototype.

---

### Restrictions on Historical Log tables

The Historical Log tables (Alarms, Drivers and SysMsgs) created with the 10.x PowerHMI version are no more compatible with the 11 version since the tables structure for 11 version includes two additional fields. Moving to 11 version these tables will be re-created, loosing all recorded data.

---

### Restrictions on Page Change commands

The "back history" screens list used for "Close and Return back" command is not available. The first time the "Close and Return back" commands is executed, the supervision goes back to the previous screen, the second time it goes back to the start screen.

---

### Restrictions on String Table

The columns of the String Table do not support neither the properties for the Reference Style (Style Source Container) nor the properties for Text -To-Speech ( TTS Engine, TTS Volume, TTS Rate. It is not possible to change the TTS engine for the Text-To-Speech function on the project language change.

---

## Movicon BA Restrictions (Buidling Automation)

The Movicon BA version (Building Automation & Domotics) has some restrictions in respect the complete version of Movicon. A list reporting these restrictions is shown below.

**Movicon BA installs and supports only the graphic symbols library explicitly designed for Building Automation, and only the Communication Drivers explicitly designed for Building Automation' devices.**

Here listed are some specific BA Drivers:

EIB/Konnex  
Elkron  
Notifier E-Net  
Guardall  
CEI-ABI (EL.MO.)  
CEI-ABI (Notifier)  
Duemmegi Contatto  
Intermod  
OVA  
LON (being prepared, available an OPC Server)



**All the specific BA drivers require an Activation Code to unlock the communication runtime.**

**The updated list of BA drivers is available from the Progea website or you can contact the Progea sales offices or distributors.**

Communication, using Movicon BA, can be also established using the following:

- **OPC**
- **Modbus RTU**
- **Modbus TCP-IP**
- **ODBC RealTime**



## 2. Important Information

---

### Assumptions

---

All the information provided in the Movicon documentation is based on the assumption that:

- All the illustrations in this manual refer to the last Movicon version with Windows™XP/Vista platform
- Windows™ XP/Vista refer to the trademarks of Microsoft inc
- Ms Access™ and SQL Server™ refer to the Microsoft inc. registered products
- Crystal Report refers to the product registered by Business Objects - Seagate, version 10 or later
- Movicon™ refers to the supervision system developed by Progea and is protected by the international Copyright
- Any other product or trademark cited is registered or protected by their relative owners
- Windows is installed in the system. For information about this installation please consult the relevant chapters in the Windows 'Introduction Guide'
- A mouse had been installed. If a mouse is not being used please check the keyboard equivalents
- All the information contained in this manual is subject to change without prior notification due to system updating

### Conventions

---

Movicon exploits the Windows standard techniques such as Copy, Paste, Drag, Multi-selecting etc., for managing projects.

This documentation assumes that you are:

- Familiar with the appropriate operating system working environment.
- Knowledgeable of how to use of a mouse, Windows menus, select options, and accessing online Help.
- Experienced with a programming or macro language. For best results, you should have an understanding of programming concepts such as variables, statements, functions and methods.
- Experienced with automation systems that the product is referred to



**Important:** The combo keys indicated in this guide, are to be considered valid provided they are not being used by in the project by the Movicon Shortcuts, i.e. the plants keyboards commands. If used in the project, the plant commands keys have precedence over the system keys.

For further information, please consult the chapter on Shortcut Resource in this guide.

# Copyrights

---

Movicon© is a software product and a copyright by Progea. All rights are reserved.  
The brand Movicon™ is a registered trademark of Progea.

Progea  
Via S. Anna 88/E  
41100 Modena, Italy

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of Progea.

## Warning

---

**This software is not guarantee for a level of reliability suitable for use in or in connection with surgical, nuclear, military or other critical plants and in any life support systems whose failure to perform can reasonably be expected to cause significant injury to a human.**

In any application, including the above, reliability of operation of the software products can be impaired by adverse factors, including but not limited to fluctuations in electrical power supply, computer hardware malfunctions, computer operating system software fitness, fitness of compilers and development software used to develop an application, installation errors, software and hardware compatibility problems, malfunctions or failures of electronic monitoring or control devices, transient failures of electronic systems (hardware and/or software), unanticipated uses or misuses, or errors on the part of the user or applications designer (adverse factors such as these are hereafter collectively termed "system failures"). Any application where a system failure would create a risk of harm to property or persons (including the risk of bodily injury and death) should not be reliant solely upon one form of electronic system due to the risk of system failure. To avoid damage, injury, or death, the user or application designer must take reasonably prudent steps to protect against system failures, including but not limited to back-up or shut down mechanisms. Because each end-user system is customized and differs from Progea' testing platforms and because a user or application designer may use Progea products in combination with other products in a manner not evaluated or contemplated by Progea, the user or application designer is ultimately responsible for verifying and validating the suitability of Progea products whenever Progea products are incorporated in a system or application, including, without limitation, the appropriate design, process and safety level of such system or application.

### **Disclaimer of All Warranties**

- PROGEA DISCLAIMS ANY AND ALL WARRANTIES WITH RESPECT TO MOVICON AND THE RELATED DOCUMENTATION, WHETHER EXPRESS OR IMPLIED, INCLUDING SPECIFICALLY THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A GENERAL OR PARTICULAR PURPOSE.
- MOVICON AND THE RELATED DOCUMENTATION ARE PROVIDED "AS IS," AND YOUR COMPANY UNDERSTANDS THAT IT ASSUMES ALL RISKS OF THEIR USE, QUALITY, AND PERFORMANCE.

### **Disclaimer of Liability**

- YOUR COMPANY AGREES AND ACKNOWLEDGES THAT PROGEA SHALL HAVE NO LIABILITY WHATSOEVER TO YOUR COMPANY FOR ANY PROBLEMS IN OR CAUSED BY MOVICON OR THE RELATED DOCUMENTATION, WHETHER DIRECT, INDIRECT, INCIDENTAL, SPECIAL, OR CONSEQUENTIAL (INCLUDING LOSS OF PROFITS).

# Windows VISTA Support

---

Movicon 11 has been expressly designed for Windows Vista™ and it passed all the tests required for "Works with Windows Vista" Logo certification:



For further information please read this Microsoft documentation:

Some modifications have been made to Movicon which consent the product to run the following operations even when the UAC is active in the Windows VISTA operating system:

- Movicon registration in the ROT (Running Object Table) is done during the product's installation phase, as well as the project startup.
- 'Alarm Dispatcher registration in the ROT (Running Object Table) is done during the product's installation phase, as well as the project startup.
- Icons associated to the project files are registered during installation, in addition to opening the project in development mode.
- The Movicon service is installed/uninstalled using the "RegService.exe" external tool. This tool is used in automatic mode when needed, and it is completely transparent to users.
- The Movicon OPC Server is registered/unregistered using the "RegOPCServer.exe" external tool. This tool is used in automatic mode when needed, and it is completely transparent to users. The automatic registration and unregistration of the OPC Server at application startup or termination remains impossible when the UAC is active in Windows VISTA.



**Important note! At the date of product release the use of the graphic interface running Movicon 11 as Windows Vista service is still not supported. Please contact technical support for further information.**



## Technical Support

---

Progea supplies a Technical Support Service very fast and highly efficient. The Technical Support Team is at customers disposal, according to service rules, for all customers having purchased at least one Editor licence and having registered as Movicon users.

Sending back the Registration form or registering the product on-line, Progea assures the customer service technical support and the product warranty according to the use licence agreement and the standard procedures.

All technical support information must be obtained from the dealer where this product was purchased. For any further information on this product or the technical support service, you can contact Progea, the product developers, by telephoning the numbers either on the registration form or the telephone assistance contract or in the "Readme.txt" file. Always quote the product's serial number when making enquiries.

### Web Support Center

The registered users can access Technical Support Web Service. Please visit our site to see Support and Services offered by Progea. And while you are there you can take a look at the BugBase, Knowledge base, Examples, FAQs, the last Build available, support centres and the Solution Providers. You can also fill in a form to let us know of any suggestions you may have.

### Support via E-mail

The registered users can also access the Technical Support Web Service by sending an e-mail to the Support Team. Each technical support request is traced and recorded in the Technical Support Database and managed as described in support section in Progea web site.

To get technical information quickly and easily via e-mail, 24 hours a day, seven days a week, send us your technical questions to [support@progea.com](mailto:support@progea.com) or your commercial requests to [sales@progea.com](mailto:sales@progea.com).

### Phone Support

The access to phone assistance service (Hot Line Support) is reserved to customers which subscribe the Technical Support contract. Please check in support section in Progea web site how to access the Hot Line support service.

If you like to subscribe a **Technical Support contract** and taking advantage of its benefits, please contact Progea sales offices at [sales@progea.com](mailto:sales@progea.com) or contact your local dealer.



Note: For a more efficient technical support, we suggest to use the "Support Inquiry" form available, support section.

The use of this form allows to better define the problem and to quickly find a solution, for a better customer service.

## Web Support Center

---

The web site for the Progea support is available at the address, on the Support section ("Web Support" link).

The access to this page require the user registration (reserved access).

A WebHelp is available on the Web Support Center page where you can check out the BugBase, Knowledgebase, the examples, FAQs and the last product Build available. This tool, which is constantly kept updated by Progea, is particularly useful to the user who will be able to see a series of examples and answers to the most frequently asked questions by clients. In addition to this there is a list of fixed bugs from different versions and information on the new items introduced.

# Updates and BugBase

---

The Movicon software is periodically updated with the release of new updating Builds by the Progea 'Research and Development' team which can be downloaded from the site free of charge.

You can also access the download page on the website to check out the last **release** available and the reasons why it was released.

- You should always check, when the system is not running correctly, whether our problem has already been put right in the **bugbase**, where there is a historical list of all the problems found so far and resolved.

The solution to a problem or improvements involves the unavoidable release of a build, which is available and can be downloaded from the Progea website free of charge.

## Updates

---

Below the Update and Upgrade concepts will be defined, referring as always to the version number of the product purchased.

As well as the 'splash' at the program's start the version number can also be checked out through the information window accessed with the "**Information On...**" command from the '**Help**' ('?') menu.

The first two numbers refer to the program version, the next three numbers refer to the Build.

For instance, version 11.0.1008 means: Version 11.0, Build 1008.

<b>Update</b>	Updates are the patches updating one build to another within the same program version. The Build change is always free of charge.
<b>Upgrade</b>	Upgrades are all changes which upgrade one Movicon version to another. Not only is the software upgraded but also the dongle and the set of manuals that go with it. The upgrade can always be purchased on request.



**Upgrading a project to a next build does not always maintain compatibility with the previous build. Before upgrading, we suggest you to check the eventual loosing of compatibility with the past. By consulting the Web Support Center, Bugbase section, you can check any eventual incompatibility list.**

## Bugbase

---

The bugbase is the program's historical archive containing a chronological report of all the version releases and builds with explanations of the solved problems and the new items inserted.

When installing new updates you should make it a habit to also read the bugbase to check-out the new items inserted, as regards to the previous versions, or to look over the resolved problems in order to save time or asking for technical support.

The Bugbase is available online at the Progea website on the **Support Center** page, so that users can go over the problems already taken care of and decide whether to they want to update or not by executing the free download.



## 3. Before Starting

---

Movicon is a software platform to create and run projects for the supervision and control any kind of automation system.

Movicon's main task is to save you time in creating supervision projects by providing you with all the necessary functionalities in resources and object libraries, prebuilt to suit any purpose.

This product has two working modes: **Development** and **Runtime**

<b>Development</b>	The development mode allows you to edit projects. In this mode, the system permits you to create or modify project files by means of inserting resources and to configure them using properties.
<b>Runtime</b>	The Runtime mode lets you run the project. In this mode the user can view data and use commands predisposed by the programmer.

Nearly all the project's documentation refers to using Movicon in "**Development**" mode with illustrated techniques for creating powerful supervision projects according to the requirements that need to be made available to the supervisor users in "runtime".

### What you can do with Movicon

---

Any necessity relating to creating automation software for vision and control, whether sophisticated supervision systems (Scada) or simple operator interface systems (HMI) can be easily created by using Movicon.

The editing techniques have been conceived to lighten the work load and reduce developing. Naturally, as for all development platforms, the optimal product use must take into consideration all the best techniques to maximize profits and optimize results.

These brief tips do not have the aim of planning you a good user interface, which is the job of the programmer based on the objectives and results required of and by client specifications, but to give you a guideline and better understanding of the techniques used and functionalities of the system before starting to use Movicon.

### Let's look at the Workspace

---

First of all, start up Movicon and briefly observe and study the workspace. This operating environment has left various tools at your disposal which, when used correctly, allow you to speed up your work.

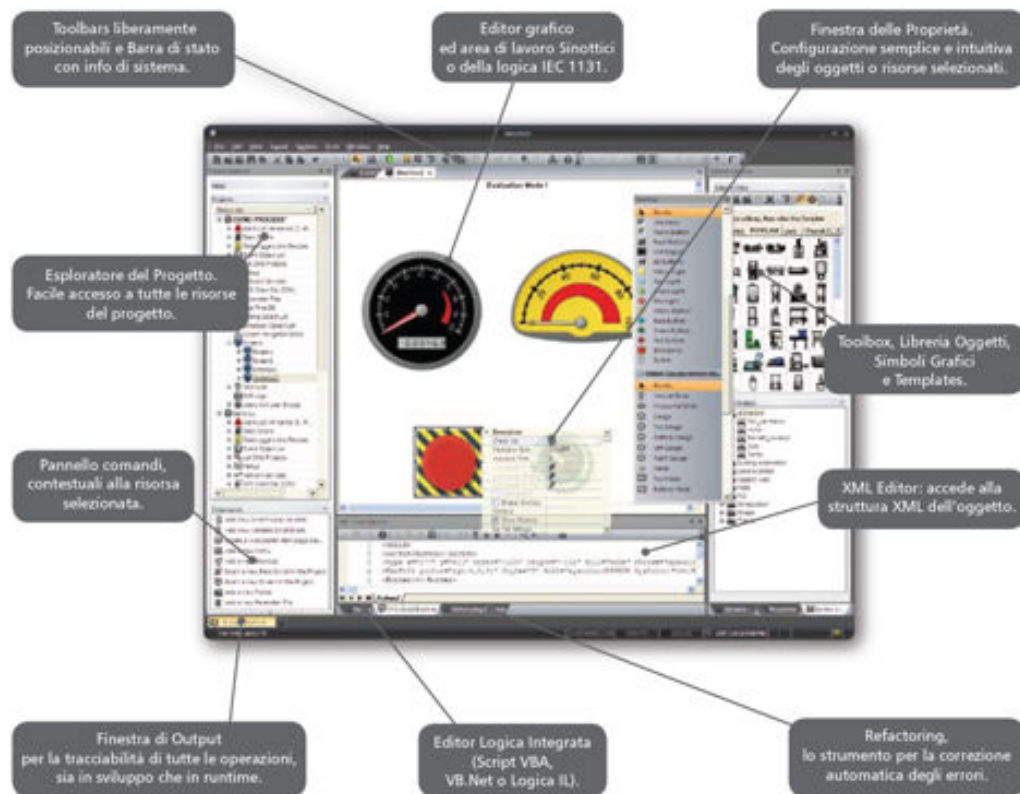
The main tools are:

Project Explorer Window: displays all the project **resources** in a tree structure

Properties Window: displays all the **properties** which can be configured for the selected object or resource

Objects Window: Displays the Toolbox with graphics **object** libraries which can be inserted on screens

Symbols Library Window: Displays the graphic **symbol** libraries



### Working in "Easy" mode

For those who are beginners, the object properties windows may seem quite daunting in how many there are available for each object. As time passes you will get to appreciate the many properties which permit you to completely configure all the functionalities reducing the need to resort to codes. However, at the beginning it would be more helpful to start working in 'Easy Mode' where the properties windows hide the more sophisticated and less used ones, reducing the availability of functions and simplifying the initial approach.

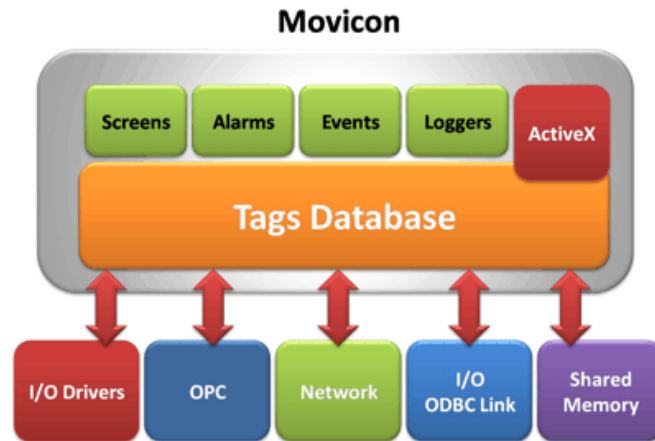
The display change is done by using the  button from the toolbar of the same window.

Linked arguments:  
Project Structure

### Let's start with the Variables from the RT Database

The Supervisor's main objectives to communicate with the field devices and provide information. The dynamic information is called 'Variables' or Tags, which reside in the Real Time DB resource. The communication drivers are setup in this resource and permit data exchange with the devices. This data is associated to the variables defined in the Variable List(Tags).

Real Time DB: Project resource which contains **driver** settings and the list of Tags Variable List(Tags): Lists all the project's **global** variables. At least the name, type and the **dynamic** device address must be set in the properties of each Tag. When dynamic addresses are not associated, the variables will be "internal" and not counted by the license.



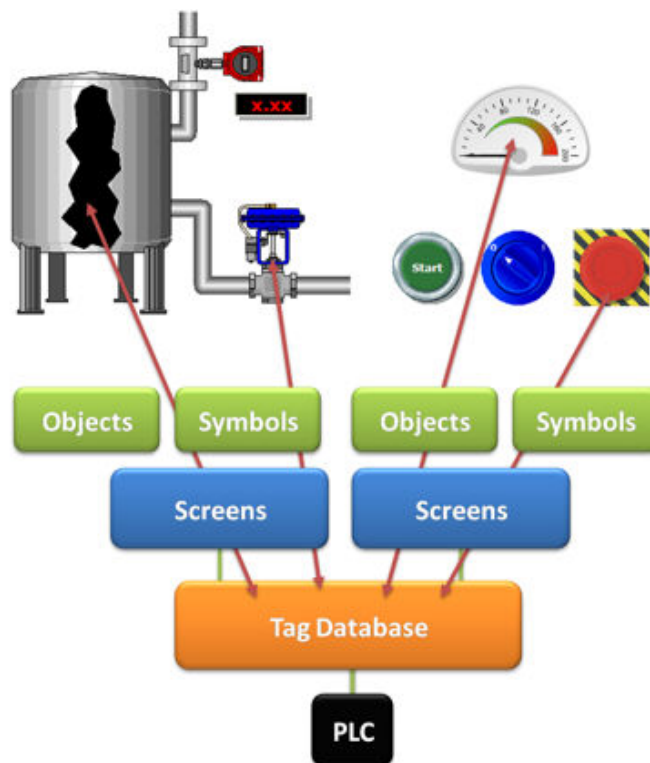
## The Graphic Interface

After having observed where the variables, these variables will contain dynamic information to be associated to the various supervisor functionalities. The Graphic Interface is one of the main functionalities, where you can create graphical displays known as video pages or Screen Resources, which are inserted into the project "Screens". The **graphic tools** (objects, symbols) can then be used in the **screens** to create drawings to build the user interface. The screens reside in the "resources" group of the project structure.

Fundamentally, each graphic object inserted on screen, is made dynamic because it is associated with the variable (or variables) in its properties for graphical animations or commands desired. The command objects can, however, execute one or more commands such as change page, data settings, operations commands on alarms, reports, etc.

Linked arguments:

Graphic Interface  
How to create Graphic Interfaces  
Editing Symbol Graphics



## The Alarms

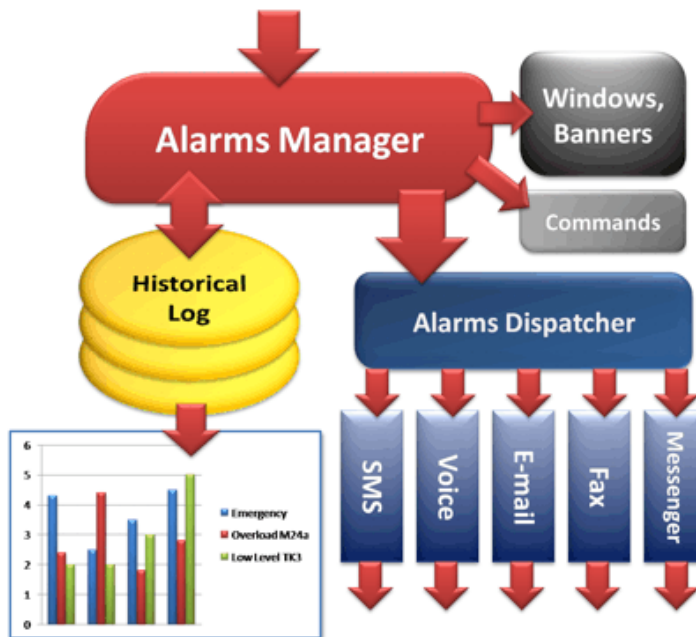
One of fundamental supervisor tasks is to manage alarms. The target is to obtain is to associate an alarm to a variable, which can have one or more intervention thresholds.

Generally, these alarms then must be filed in a log, known as the Historical Log, which can be displayed as pleased. These alarms can then be set to give notifications via SMS, E-mail, Voice etc.

The alarms are setup in the project's Alarms Resource where the 'alarm' objects receive the variables associated in their properties. The thresholds are then set and the properties relating to the text, logic conditions along with and any colors and styles.

Once set, the alarms are then managed in the project. When the condition is activated, the alarm will display in a predisposed Alarm Window, which is a graphic object from the toolbox that can be inserted in any screen.

The alarms, if not specified otherwise in their properties, are recorded on filed in the Log by the system. The stored data can be viewed through the Log Window, which is another graphic object from the toolbox which can be inserted in any screen.



Linked Arguments:

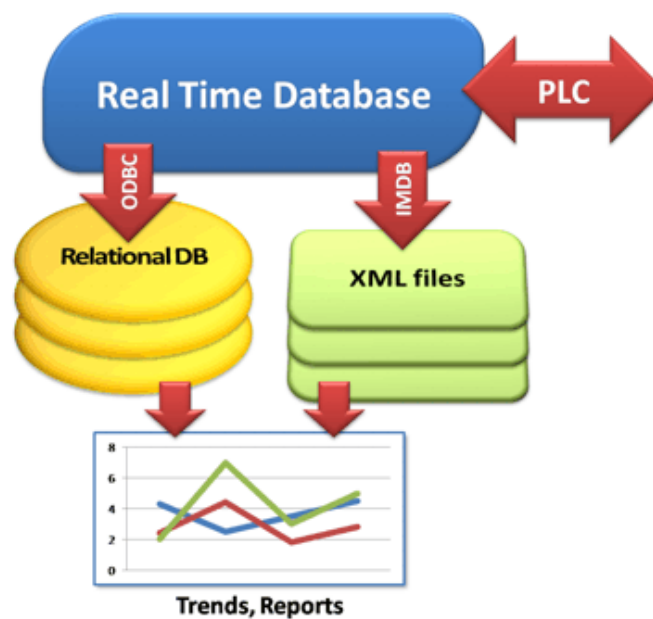
Alarms

Inserting Alarms

Historical Log

## The Historical Logs

The log recording tools constitute one of the fundamental aspects of any supervisor. In addition to historically recording events that, as indicated above, takes place through the Historical log tool, it is often necessary to record data log files containing process information. These data files can be used for the production Recipes or for recording process variable on log (Data Log). The tool to use in each case, is the Data Loggers and Recipes resource.



The DB objects can be used in both cases which manage the filing of data as needed when configured appropriately. Recorded data from the Data Loggers and Recipes may be displayed for the following reasons:

- Recipes
- DataLogger Window
- The Grid
- Chart Objects
- Trend
- Report (Crystal Report)

The desired Data Logger can be associated to any one of the above mentioned graphic objects, being a historical data base managed by the supervisor. The graphic objects display historically logged data according to the contents in the filed managed by the Data Logger objects.

## The Logics

Even though the job of a supervision platform is to avoid, as much as possible, the use of codes (to reduce development times), it may be necessary to resort them nevertheless for executing calculations, logic functions whether for customizing the object's or resource function to adapt them to the specifications required. In this case, the logics permits you to get the function type you want, even when not provided in the object's properties. Movicon provides you with three types of languages to create control logics:

VBA Language (Visual Basic for Application)

- Basic Scripts as Resources
- VBA™ Basic Script in Object Properties
- VBA™ Basic Scripts in Object code

PLC-like Language

- IL Logic (Instructions List)

Synapses (graphic object orientated language, based on VBA)



## Other Objects and Resources

---

Movicon offers many other functionalities, such as menus, accelerators (keyboard) schedulers, event objects, change language, Users and password management, networking, OPC, modem, etc.. Each functionality is described in the appropriate chapter. In this context, the aim was to give you a general view of the main functionalities and basic concepts of use.

## Microsoft Windows Certified

---

Movicon, in full respect for the standards required by Microsoft for Windows™ compatibility, completely supports the standard resource usage and management techniques.

Movicon has been certified by Microsoft Logo Program "Designed for Windows XP".



The Microsoft standard techniques supported refer to:

- Editing Techniques
- Drag & Drop
- Right mouse key
- Color selections
- File, font, printout selections

The Microsoft standards supported in managing system and other applications refer to:

- ODBC Support
- SQL Support
- OLE2, OLE2 Automation Support
- VBA Compatible and ADO Support
- OCX - ActiveX Support

This part of the guide will briefly deal with the Movicon editing and usage techniques being the Windows standard techniques.

The less advanced user should refer to the Windows User's Manual for further information on these techniques.

# Project Programming Limits

---

There are some limits in Movicon programming and the components you can use (ie. Database, ODBC, etc) that should be taken into account while designing projects. These limits have been listed in the tables below considering that:



1. **These recommended limits are not constraints but suggestions based on tests run by the builders. It is totally the user's responsibility to decide what's best for their project according to hardware resources being used and the total amount of resources in use.**
2. **The resource limits refer to projects run on Personal Computer with at least WinXP PRO and CPU Pentium 4 configuration with at least 2 GB RAM.**
3. **Future programming limits may be necessary, even though not stated in the following table or indicated the product's documentation, caused by restrictions deriving from circumstances or a combination of circumstances internal or external to the project.**

Real Time DB	
Variable Names	<ul style="list-style-type: none"><li>• Variable names cannot contain special chars other than "_\"</li><li>• Variable names cannot begin with a special char or a number</li><li>• Variable names can be up to 64 chars long</li></ul>
Structure Variable Size	Structure variables cannot exceed sizes of 32767 bytes. If this limit is exceeded, a message will appear in the log both in design and runtime mode.
Maximum number of Variables	<p>Even though Movicon does not impose a maximum limit on variables declared in projects, the recommended limit however would be a number of 100,000 variables (Tags) declared per project for Windows Vista/XP.</p> <p>Projects more variables than this recommended limit can be managed based on the hardware sources being used or the total number of resources being used in the project.</p> <p>Rather complex projects can always be distributed in additional projects declared as "Child Projects", both locally or physically in different PCs in net.</p>

Screen Objects	
Gradual filling	The "Pipe" object does not support Gradual Fillings.
Expressions	Object which use VBA expressions cannot use Local Variables within the expression.
Maximum number of objects on screen	<p>Even though Movicon does not impose a maximum limit on the number of graphical objects which can be inserted in each screen, the recommended limit is 5000 vector segments declared per screen in project for Windows Vista/XP.</p> <p>Projects with higher quantities can be managed in function with the hardware resources being used.</p>

<b>Alarms</b>	
Maximum number of Alarms	Even though Movicon does not impose a maximum limit on the number of alarms declared in one project, the recommended number is 10,000 simple alarms per project for Windows Vista/XP. Projects with higher numbers can be managed in function with the hardware resources being used or the total number of resources being used in the project. Rather complex projects can always be distributed in additional projects declared as "Child Projects", both locally or physically in different PCs in net.

<b>Historicals and ODBC Links</b>	
DSN Name	Some Databases, such as the SQL Server does not support spaces written in the name of the DSN. Therefore it would be best not to insert spaces in project names to avoid the automatic Database creation for Historicals failures when using the SQL Server for instance.
Historical Column Names	Some Databases, such as the SQL Server CE do not support spaces written in Database column names. Therefore it would be best not to insert spaces in the names of the columns of Historical columns, such as those used in the DataLogger, Recipes, etc.
Recording Frequencies	Even though data sample frequency depends on many factors external to the project, it may be useful to keep recording frequencies not less than 50 milliseconds.

<b>VBA Script and Logic</b>	
Maximum number of scripts	Even though Movicon does not impose any limits in the number of scripts you can declare in one project, the recommended limit would be 1000 scripts per project for Windows Vista/XP, comprising of 16 executable at the same time (multithread). Projects with more than this can be managed in function with the hardware resources being used or the total number of resources being used in the project. Rather complex projects can always be distributed in additional projects declared as "Child Projects", both locally or physically in different PCs in net.
Maximum number code line in IL logic	Even though Movicon does not impose any limits in the number of code lines which can be written in the General Logic, but the recommended number is 10,000 lines per project for Windows Vista/XP.

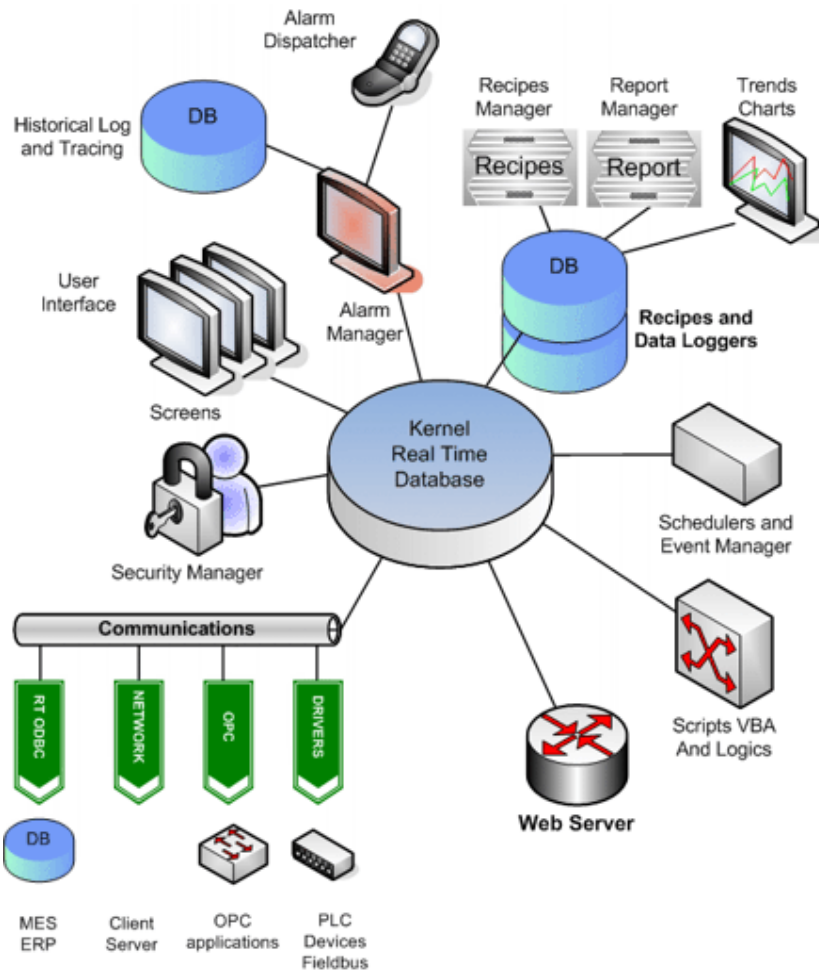
<b>Clients</b>	
Maximum number of variables connected in networking to Client stations	Even though Movicon does not impose any limits in the maximum number of variable connect in networking, it would be best to stick to not more than 10,000 connected per project for Windows Vista/XP. Be reminded that Movicon manages notifications on Event. Rather complex projects can always be distributed in additional projects declared as "Child Projects", both locally or physically in different PCs in net.

Maximum number of Client Stations connect to a Movicon Server	Even though Movicon does not impose any limits in the maximum number of client stations you can connect to a Movicon Server in net, it is recommended not to exceed the number or 100 client stations to one Server in Windows Vista/XP. Higher numbers of clients stations may be acceptable in function with the resources available and the type of network used, but Progea cannot guarantee this.
Maximum number of Web Client Stations connected to one Movicon	The number of Web Client stations connected to one server has a fixed limit of 100 web client users connected simultaneously to one Movicon Server in Windows Vista/XP. A higher number of Web Client stations may be acceptable in function with the resources available on the server, the project run on the server and the network typology used, but Progea cannot guarantee this.



## 4. Projects Structure

The Movicon Projects are composed of diverse component groups (known also as project Resources), such as the Real Time Data DB, the Resources, The Data Loggers, etc. These groups of components are displayed in a tree structure in the "Project Explorer" window and are described in full detail in the appropriately dedicated sections.



This figure illustrates the Movicon project structure in a block diagram.

The "Project Explorer" window is the main window used in programming a Movicon project.

This window is displayed straight away upon opening a project and its function is to report the Resources and their Groups of Components, contained within, in a tree structure. Each single object can easily be selected from this tree structure to carry out any editing or settings through the corresponding "Properties Window".

# A New Project

---

An application project is built with folders and files created by Movicon and programmers. When the command for creating a new project (Menu **File->New**) is executed, Movicon opens a dialog window through which three options can be executed, when the '**New**', '**Existing**' and '**Recent**' Tab are selected:

1. create **New** project
2. opens **Existing** project
3. opens a project in the list of **Recent** projects

Obviously a new project is created only in the first case, while an already created project will be opened in the other two situations.

The same dialog window will also be opened when the project's open command ((Menu **"File->Open"**)) is executed with a difference that only the options in point 2 and 3 will appear.

## New

---

When selecting the 'New' Tab, you will be asked to choose which project type you wish to create and, above all, which destination software platform the project is to be executed on.

The choices are:

1. **Movicon Project for Win32 platform**
2. **Movicon Project for Terminals (j2se)**
3. **Movicon Project for WinCE platform**
4. **Movicon Project for Mobile phones (j2me)**

After having chosen the platform desired another dialog window will display through which the projects start settings are defined. These settings, described in the relative sections, concern:

- Project Name**
- Users**
- Communication Drivers**
- Screens (Pages)**
- DataBase Settings (ODBC)**
- Data Loggers and Recipe Settings (ODBC)**
- Alarms Settings**

## Existing

---

When selecting the 'Existing' Tab the usual Windows' window will display to execute the browser of the local computer or network resources for selecting the project you wish to open for programming.

In this case you have to select a Movicon project file with the ".movprj" extension.

## Recent

---

When selecting the 'Recent' Tab the list of the last projects opened, in chronological time order from most recent to the oldest, will be displayed. Select the project you wish to open for programming from this list.

## Project Name

---

When creating a new project you can define the 'Project Name' settings to assign the Project with a name, its location and its file encoding type. It is obligatory to enter a project name in order to continue programming. These settings can also be changed afterwards during the developing phase.

**Name**

The Project's name is entered here. Specify the name without extensions only.

**Folder**

The path and the folder where the Project is to be saved is entered here. You can also use the Browse button ("...") on the right.

Movicon will propose the 'My Documents' folder as the path for the user logged in Windows for default, adding the 'Movicon Projects' folder and the folder with the same Name as the Project's. To make things clearer let's suppose that the Log-on has been carried out by 'Administrator' user and the new project has already been named 'Project1', the path and the folder proposed by Movicon will then be:

**C:\Documents and Settings\Administrator\My Documents\...**

Any part of this path can be changed due forth.

**Crypt Core Project File**

When this option is checked the Movicon Project file will be saved in encrypted mode, and therefore cannot be read with other editors. This setting can be changed after the project has been created, through the "Crypted Project" properties accessed from the 'General Project Settings'.

**Crypt all Project Resource Files**

When this option is checked the Movicon Project resource files will be saved in encrypted mode, and therefore cannot be read with other editors. This setting can be changed after the project has been created through the "Crypted Project Resources" properties accessed from the Project's 'General Settings'.

**Compress all the files**

When this option is checked the Movicon Project will be saved in compressed format. This setting can be change after the project has been created through the "Zipped Project" properties accessed from the Project's 'General Settings'.

**Encode using Unicode UTF-16**

When this option is checked the Movicon Project will be saved in Unicode UTF-16 format. This setting can be change after the project has been created through the "Unicode Project" properties accessed from the Project's 'General Settings'.

## Users

---

When creating the new project you can define the 'User' settings to create default users and groups and define whether or not to activate the password management in the Project. These settings can also be modified afterwards, during the project developing phase.

**Password Protected Project**

When this option is enabled the project's password protection will be activated. When this protection is enabled only users with Developer rights (level 1024) can open the project in developing mode. This setting can be changed after the project has been created through the "Password Protected Project" property which is accessed from the 'Users and User Group General Properties'.



When activating the 'Password Protected Project' option you must also fill in the 'Developer Name', 'Developer Password' and 're-type Developer Password' text boxes.

**Developer Name**

This edit box is activated only if the 'Password Protect Project' setting has been enabled and must be used for entering the Project Developer's user name. Movicon will automatically assign the user with a Developer Level. When the project has been created you can add other users with Developer Levels through the 'User & User Groups' management to allow the project to be opened to and by other users.





Alphanumeric and 'Case Sensitive' characters can be inserted into this field and discriminates between lowercase letters and uppercase ones.  
The minimum length for the User Name set for default by Movicon is four characters. We advise you not to go lower than this number for normative and security reasons.

#### **Developer Password**

This edit box is activated only if the 'Password Protect Project' setting has been enabled and must be used for entering the Project Developer user's password.



Alphanumeric and 'Case Sensitive' characters can be inserted into this field and discriminates between lowercase letters and uppercase ones.  
The minimum length for the User Name set for default by Movicon is four characters. We advise you not to go lower than this number for normative and security reasons.

#### **Re-type Developer Password**

This edit box is activated only if the 'Password Protect Project' setting has been enabled and must be used for re-entering the Project Developer user's password to confirm whether this has been typed correctly. The characters are inserted in encrypted mode.

#### **Enable Password Mng**

Enabling this option will activate the project's password management during Runtime phase. This setting can be changed after the project has been created through the "Enable Password Manager" property which is accessed from the 'Users and User Group General Properties'.

#### **Create Default User Groups**

When enabling this option Movicon will insert four default User Groups in the 'Users & User Groups' in the 'Project Explorer' window: "**Developers**", "**Administrators**", "**Users**" and "**Guests**". Apart from the 'Developer' group the other three use the usual O.S. group modality. When the project has been created other groups can be added or the existing ones can be modified through the 'Users & User Groups' management.

#### **Create Users from Windows Users -> Server Name**

When enabling this option the users belonging to the Operating System dominion will be inserted into the 'Users & User Groups' in the 'Project Explorer' window by Movicon. These Users will automatically be inserted in the relative "**Administrators**", "**Users**" and "**Guests**" groups if the 'Create Default User Groups' option has been enabled. If a name of a Server Network has been specified in the edit box on the right the Users of that Server will be acquired. After the project has been created other users can be added or existing ones can be modified or moved to other Groups through the 'Users & User Groups' management.

#### **Enable Runtime Users' changes**

Enabling this option will activate the Runtime Users management where it will be possible to add, change and cancel new users during project Runtime. This setting can be changed after the project has been created through the "Enable Runtime users" property which is accessed from the 'Users and User Groups General Properties'.

#### **Enable Windows User Login**

When enabling this option it will become possible to share users from the operating system domain or Windows Server in the applied project. This setting can be changed after the project has been created, through the "Enable Windows Users" property which is accessed from the Windows Users and User Groups Property'.

#### **Enable CRF-21 Settings**

When enabling this option any editing done to the User settings will be managed in conformance with the "**FDA21 CFR Part 11**" normative. This setting can be changed after the project has been created through the "Enable CRF21- Part 11 Settings" property which is accessed from the 'Users and User Groups General Properties'.

## Communication Drivers

---

When creating a new project you can define the 'Add Communication Driver' settings to enable one or more communication Drivers to be used in the Project. These settings can then be also edited afterwards, during the project developing phase, through the 'Real Time DB -> List Available Communication Driver ' in the Project Explorer' window.

Only the Communication Drivers to be used can be selected in this phase. Their settings can be done during the project developing phase.

## Screens

---

When creating a new project you can defined the 'Screens' settings to create a certain number of screens with automatically set sizes. The screens in question can then be modified afterwards during the project developing phase.

### Create Nr. Screens

The number of Screens, to be inserted in the "Screens" group in the 'Project Explorer' window to create the project, is entered in this edit box. The screens in question can be modified afterwards during the project developing phase.

### Add Screen Caption

When this box is checked, a text box with the Screen's name will be entered into the Screens which were inserted to create the project. This means that each screen will be identified with a header. Since they are realised with vectorial drawings, the header can be changed or removed during the programming stage.

### Add Screen Navigation Bar

Enabling this option, apart from the Screens set in the 'Create N. Sub Screens', another Screen will be inserted in the project being created which will appear as a command bar (reduced height) to contain as many buttons as the screens set in the 'all the button which have been 'Create N. Sub Screens'.

Each button will open the Screen it is related to (Page1 Button = opens Screen page 1). In addition to this an embedded Footer Screen will be inserted in each page so that the bar of command buttons is available in all of them. This results in having a determined number of screen pages contained in a command bar to execute page changes. All the pages and command can be changed and customized during the programming phase.

### Default Screen Width

The default width in pixels with which the Screen is to be created is entered in this edit box. This setting can be changed after the project has been created through the "CX" property which is accessed from the 'General Screen Properties'.

### Default Screen Height

The default height in pixels with which the Screen is to be created with is entered in this edit box. This setting can be changed after the project has been created through the "CY" property which accessed from the 'General Screen Properties'.

### Default Color

This selection box is used for assigning the Screen background color.

For further information please refer to the paragraph on "Colour Threshold Settings in Drawings and Controls".

## DataBase Settings (ODBC)

---

When creating new projects you can set the "DataBase Settings (ODBC)" which Movicon will use for default for recording data inherent to the Historical Log and the Variable Trace.



**CAUTION:** The creation of the ODBC link is not selected for default to permit you to choose whether to use ODBC and IMDB straight away or at another time afterwards. When neither of the two are chosen, Movicon will:

1. Automatically create the ODBC link at the first startup of a project run for Windows (Win32) projects. Movicon will attempt to create files and tables in MsSQL Server format, which will need at least the pre-installation and activation of the MSDE. Otherwise a MsAccess database will be created as an alternative
2. Set for default the historical log management in the IMDB for WinCE projects which is deemed more suitable for embedded or mobile devices



**WARNING!** Although Microsoft also releases a ODBC driver for Excel™, this is not however a DataBase, but an electronic worksheet. It is for this reason that Excel™ does not support all the SQL commands compatible with the most common Databases and therefore it is impossible to use Excel™ as if it were a DataBase. The Movicon resources such as the Data Loggers, Recipes, Historical Logs, Variable Trace etc., can therefore not be managed with a ODBC link through a Excel™ file.

### Create Historical Log ODBC DSN

When this is checked Movicon will create the ODBC Link the DataBase and the Tables inherent to the recording of data for the Historical Log. The formats which can be selected are:

- **MS Access®:** a DataBase is created in Access format
- **MS Excel®:** a worksheet is created in Excel format
- **MS SQL Server®:** a Database is created in SQL Server format
- **MySQL Server®:** a Database is created in MySQL Server format

#### **ODBC link**

The ODBC link is always created by using the following name:

**ProjectName\_HisLog**

#### **DataBase**

As regards to the DataBase files there are a few differences according to the driver selected:

**MS Access:** the DataBase will have the name "ProjectName\_HisLog.mdb" and will be placed in the project's "LOGS" folder

**MS Excel:** the worksheet will have the name "ProjectName\_HisLog.xls" and will be positioned in the project's "LOGS" folder

**MS SQL Server:** the DataBase will have the name "ProjectName" and will be inserted in the MSDE driver's installation "..\MSSQL\Data" folder

**MySQL Server:** the DataBase used is the one already set by MySQL with the name 'Test', found in the MYSQL Server engine installation "..\mysql\data" folder

#### **Tables**

The names of the Tables inserted in the DataBase, or worksheets if Excel, are:

**Alarms:** table containing alarm messages

**Drivers:** table containing driver error or warning messages

**SysMsg:** table containing system error or warning messages

### Create Trace DB ODBC DSN

When this is checked Movicon will create the ODBC link and the DataBase inherent to the recording of data for the Movicon Variable Trace. The relative tables will be created after through the appropriate "Variable Trace Options Proprieties". The formats which can be selected as default are:

- **MS Access®:** a DataBase is created in Access format
- **MS Excel®:** a worksheet is created in Excel format
- **MS SQL Server®:** a Database is created in SQL Server format
- **MySQL Server®:** a Database is created in MySQL Server format

### **ODBC link**

The ODBC link is always created by using the following name:

**ProjectName\_TraceDB**

### **DataBase**

As regards to the DataBase files there are a few differences according to the driver selected:

**MS Access:** the DataBase will have the name "ProjectName\_TraceDB.mdb" and will be placed in the project's "DATA" folder

**MS Excel:** the worksheet will have the name "ProjectName\_TraceDB.xls" and will be positioned in the project's "DATA" folder

**MS SQL Server:** the DataBase will have the name "ProjectName" and will be inserted in the MSDE driver's installation "..\MSSQL\Data" folder

**MySQL Server:** the DataBase used is the one already set by MySQL with the name 'Test', found in the MYSQL Server engine installation "..\mysql\data" folder

### **Tables**

The names of the Tables inserted internal the DataBase, or the names of the worksheets, are defined with the same name of the variable they represent. i.e. if the trace option is enabled for the variable called "VAR00001", a Table called "VAR00001" will be created in the DataBase for every variable enabled with the Trace.

## Data Loggers and Recipe Settings (ODBC)

When creating a new project you can set the 'Data Logger and Recipe Settings (ODBC)' to define which ODBC link is to be used by Movicon for default to record data inherent to the project's Data Loggers and recipes.



**WARNING! Although Microsoft also releases a ODBC driver for Excel™, this is not however a Database, but an electronic worksheet. It is for this reason that Excel™ does not support all the SQL commands compactible with the most common Databases and therefore it is impossible to use Excel™ as if it were a Database. The Movicon resources such as the Data Loggers, Recipes, Historical Logs, Variable Trace etc., can therefore not be managed with a ODBC link through a Excel™ file.**

### **Create Data Loggers And Recipes DSN ODBC**

When this is enabled Movicon will create the ODBC link and the DataBase inherent to the recording of data for the Movicon Data Loggers and Recipes. The relative tables are created after through the appropriate "Create DB Table" property which can be accessed from the Database Settings common to Data Loggers and Recipes Properties'. The formats which can be selected as default are:

- **MS Access®:** a DataBase is created in Access format
- **MS Excel®:** a worksheet is created in Excel format
- **MS SQL Server®:** a Database is created in SQL Server format
- **MySQL Server®:** a Database is created in MySQL Server format

### **ODBC link**

The ODBC link is always created by using the following name:

**ProjectName\_DLR**

### **DataBase**

As regards to the DataBase files there are a few differences according to the driver selected:

**MS Access:** the DataBase will have the name "ProjectName\_DLR.mdb" and will be placed in the project's "DLOGGERS" folder

**MS Excel:** the worksheet will have the name "ProjectName\_DLR.xls" and will be positioned in the project's "DLOGGERS" folder

**MS SQL Server:** the DataBase will have the name "ProjectName" and will be inserted in the MSDE driver's installation "..\MSSQL\Data" folder

**MySQL Server:** the DataBase used is the one already set by MySQL with the name 'Test', found in the MYSQL Server engine installation "..\mysql\data" folder.

## Tables

The names of the Tables inserted internal the DataBase, or the names of the worksheets if Excel, are defined with the same name of the Data Logger or Recipe. i.e. if a Data Logger call 'Temperature' is inserted, a Table called 'Temperature' will be created in the DataBase. Therefore a table will be created for every Data Logger and Recipe inserted into the project.

### Create Data Logger Template

When this box is checked 1 to 5 Data Loggers will be automatically added to the list of the project's Data Loggers as Templates. The only difference between these Data Loggers is the recording which may be:

**every 5 seconds**  
**every 10 seconds**  
**every 30 seconds**  
**every minute**  
**every 10 minutes**

otherwise they are all the same without any associated columns. Their purpose is to be used as Templates which can be associated to variables directly, as described in the section on "Associating Data Loggers and Recipes to Variables", when diverse variables are to be recorded with the same modalities. This is a very quick way of assigning variables to DataLoggers.

The programmer then has the task of carrying out any changes to the Data Logger settings regarding the recording modalities and columns can be added to them or they can be used as standard Data Loggers and not as Templates.

## Alarms Settings

---

When creating a new project you can use the 'Alarms Settings' to create a digital and an analog alarm to be eventually used as Templates.

### Create Digital Alarm Template

When enabling this selection box a Digital Alarm will automatically be added to the project's 'Alarms List', which is an alarm with only one intervention threshold set at '1' value, and a Digital Message, being an alarm with only one intervention threshold set at value '1' with its "Support ACK" and "Support RESET" disabled. This alarm and this message can be used as Templates and associated directly to the Movicon Real Time DB variables. An alarm or message can in fact be associated to several variables, as described in the "Alarms as Template" section, in cases where the alarm or message type remains always the same and when repeated for different devices. However the alarm or message can always be configured in the normal way, associating them a variable and managing them as single alarms or messages.

### Create Analog Alarm Template

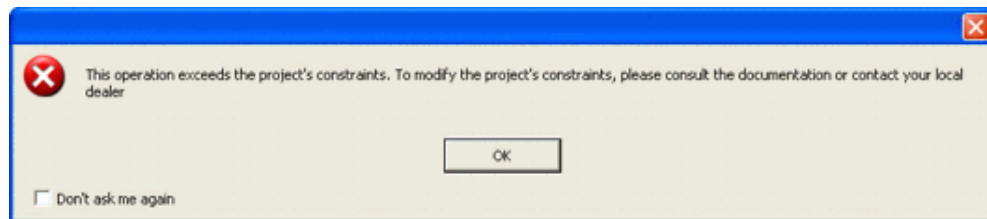
When enabling this selection box an Analog alarm will automatically added to the project's 'Alarms List', which is an alarm with more than one intervention thresholds. These alarms have four thresholds which are defined as follows:

**LowLow** (intervention for values < -100)  
**Low** (intervention for values < -90)  
**High** (intervention for values > 90)  
**HighHigh** (intervention for values > 100)

This alarm can be used as a Template to be directly associated to the variables from the Movicon Real Time DB. An alarm can also be associated to more than one variable, as described in the section on "Alarms as Template", when the same alarm type is repeated on different devices. The programmer therefore should configure the alarm to be used as template appropriately. The alarm can always be configured as a normal alarm, by associating it to a variable and managing it as a single alarm.

## Project Constraints

Each project can be set with constraints when being developed. When these constraints are exceeded the following alert window will appear:



**These constraints are managed with a xml file which has the name of the project with the ".Constraints" extension. These files contain default values which can be changed. The purpose of these constraints is to set projects with limits to stop them from getting too big for the chosen platform they are to be run on. However, it is important to keep in mind that any possible value changes made to these limits may degrade the project's performances.**

Furthermore, there are some items which consent certain information in the project structure to be displayed or hidden such as that shown in the Project Explorer Window.

The various attributes in the ".Constraints" file must be inserted in the right order so that they can be valued properly. Following this, the ".Constraints" file is proposed once again with the list of all the attributes, in order, that have been set with the default value:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<Constraints>
  <MaxNumVariables>4294967295</MaxNumVariables>
  <MaxNumScreens>4294967295</MaxNumScreens>
  <MaxNumScripts>4294967295</MaxNumScripts>
  <AllowScriptInScreen>1</AllowScriptInScreen>
  <AllowILInScreen>1</AllowILInScreen>
  <MaxNumScreenObjects>4294967295</MaxNumScreenObjects>
  <MaxScreenSizeCx>2147483647</MaxScreenSizeCx>
  <MaxScreenSizeCy>2147483647</MaxScreenSizeCy>
  <MaxDaysAgeHistoric>65535</MaxDaysAgeHistoric>
  <MaxCacheBeforeFlush>32767</MaxCacheBeforeFlush>
  <MaxNumberRecordsDL>10000</MaxNumberRecordsDL>
  <AllowGradientFill>1</AllowGradientFill>
  <EditVariables>1</EditVariables>
  <EditDrivers>1</EditDrivers>
  <EditStructures>1</EditStructures>
  <EditScreens>1</EditScreens>
  <EditScripts>1</EditScripts>
  <EditShortcuts>1</EditShortcuts>
  <EditMenus>1</EditMenus>
  <EditParameters>1</EditParameters>
  <EditUsers>1</EditUsers>
  <EditDLRs>1</EditDLRs>
  <EditAlarms>1</EditAlarms>
  <EditEvents>1</EditEvents>
  <EditSoftLogic>1</EditSoftLogic>
  <EditScreenNavigation>1</EditScreenNavigation>
```

```

    <EditSchedurs>1</EditSchedurs>
    <EditScaling>1</EditScaling>
    <EditChildProjects>1</EditChildProjects>
    <EditNetwork>1</EditNetwork>
    <EditOPCDACOM>1</EditOPCDACOM>
    <ShowExtendedBrowseTag>0</ShowExtendedBrowseTag>
</Constraints>

```

Each attribute means:

- **MaxNumVariables:** maximum number of variable that can be entered in the project's RealTimeDB
- **MaxNumScreens:** maximum number of screens that the project can have
- **MaxNumScripts:** maximum number of basic scripts that can be entered in the project
- **AllowScriptInScreen:** allows the script code to be edited in the screen or in its objects (default value is "true")
- **AllowILInScreen:** allows the IL codes to be edited in the screen or in its object (default value is "true")
- **MaxNumScreenObjects:** maximum number of objects that can be inserted on one screen. One symbol counts as one object
- **MaxScreenSizeCx:** maximum length set in pixels that a screen can be set with
- **MaxScreenSizeCy:** maximum height in pixels that a screen can be set with
- **MaxDaysAgeHistoric:** specifies the maximum project historical table age in days. This setting is used in the Historical Log tables, the variable Trace DB tables, and in Data Logger tables. This value is also used in project design mode to reduce the default age of the various historical logs, in cases where values may be more than the set limit.
- **MaxCacheBeforeFlush:** consents the maximum size of the cache to be specified before the system unloads data on files. This value represents the "Max. cache size" property of the "ODBC manager" properties group, for the project historical logs or for DataLoggers.
- **MaxNumberRecordsDL:** consents the entry of a further control on Data Loggers. Frequency values or age values that produce a total number of records higher than this limit, set through the constraints, will not be accepted. The "Tot. Nr. Records" property allows you to see the total number calculation of records estimated for a determined Data Logger. This calculation uses the age set in the historical and the sampling frequency in this formula:

$$\text{Total Nr. Records} = (\text{Data Max.Age/Sampling Frequency}) * 2$$

ie: When setting MaxNumberRecordsDL = 68400 in the constraints file, it will not be possible to change the "Data Max Age" or the Sampling Frequency if this results  $(\text{Data Max.Age/Sampling Frequency}) * 2 > 68400$ . Only the "MaxDaysAgeHistoric" value is used when creating a new project, while the **MaxNumberRecordsDL** value is controlled only when the Data Logger age or sampling frequency are being edited

- **AllowGradientFill:** when set at 0, consents all the properties, for setting the back color Gradient types of screens and objects, to be hidden
- **EditVariables:** when set at 0 the "Real TimeDB" node in the Project Explorer window will not be displayed, the "Edit Real Time DB Variables" will be disabled from the project's Edit menu (right mouse click), the "Edit Realtime DB Variable..." and "Add a New Variable (Tag)" from the Project Explorer's window's command panel will also be disabled
- **EditDrivers:** when set at 0 the 'List Com.Drivers' sub-node from the RealTime DB group in the Project Explorer window and the "Add New Comm.Driver" from the same menu of the "Real TimeDB group and the "Add New Comm.Driver" item from the Project's Explorer's command panel will also be disabled
- **EditStructures:** when set at 0 the "List Structure Prototypes" sub-node from the RealTime DB group in the Project's Explorer window and the "New Structure Definition" from the Real time DB edit menu and the "Add a new Struct Prototype" item from the Project's Explorer window's command window will both be disabled.
- **EditScreens:** When set at 0 the "Screens" node in the Project's Explorer window will not be displayed, the "Add a new Screen" from the edit menu (right mouse

click), the "Insert a new screen in the Project" item in the Project's Explorer window's command panel and the "New Screen" icon from the Toolbar will all be disabled

- **EditScripts:** when set at 0 the "Basic Script" node in the Project's Explorer window will not be displayed, the "Add a new Script" the edit menu (right mouse click), the "Insert a new Basic Script in the Project" in the Project Explorer window's command panel and the "New script" icon in the Toolbar will all be disabled
- **EditShortcuts:** when set at 0 the "Shortcuts" node in the Project Explorer window will not be displayed, the "Add a new Shortcut" item from its edit menu (right mouse click), and the "Add a new Shortcut" from the Project Explorer window's command panel and the "new shortcut" icon in the toolbar will all be disabled
- **EditMenus:** when set at 0, the "Menus" node in the Project Explorer window will not be displayed, the "Add a New Menu" item from the edit menu (right mouse click), the "Add a new Menu" from the Project Explorer window's command panel and the "New Menu" icon in the Toolbar will all be disabled
- **EditParameters:** when set at 0, the "Parameter Files" node will not display in the Project Explorer window, the "Add a new Parameter File" from the edit menu (right mouse click), the "Add a new Parameter File" item from the Project Explorer window's command panel and the "New Parameter File" icon from the Toolbar will all be disabled
- **EditUsers:** when set at 0, the "Users and User Groups" node will not be displayed in the Project Explorer window
- **EditDLRs:** when set at 0, the "DataLoggers And Recipes" node will not be displayed in the Project Explorer window
- **EditAlarms:** when set at 0, the "Alarm List()" node will not be displayed in the Project Explorer window
- **EditEvents:** when set at 0, the "Event Object List" node will not be displayed in the Project Explorer window
- **EditSoftLogic:** when set at 0, the "Soft Logic" node will not be displayed in the Project Explorer window
- **EditScreenNavigation:** when set at 0, the "Screen Navigation Editor" node will not be displayed in the Project Explorer window
- **EditSchedulers:** when set at 0, the "Scheduler Object List" node will not be displayed in the Project Explorer window
- **EditScaling:** when set at 0, the "Scaling Object List" node will not be displayed in the Project Explorer window
- **EditChildProjects:** when set at 0, the "List Child Projects" node will not be displayed in the Project Explorer window, the "New Child Project...." item in the edit menu (right mouse click), the "New Child Project...." item in the Project Explorer window's command panel will all be disabled
- **EditNetwork:** when set at 0, the "Network Services" node will not be displayed in the Project Explorer window
- **EditOPCDACOM:** when set at 0, the "OPC Client DA (COM) " node will not be displayed in the Project Explorer window
- **ShowExtendedBrowseTag:** when set at 0, allows the "OPC", 'Network' and "Comm. I/O Drivers" tabs to be hidden in the Tag Browser window. This should stop programmers from inserting dynamic addresses to variables (Fixed I/O address)

The properties that are not in this file are to be considered active, meaning as if set at 1. However, those that have been inserted will be managed according to the value entered in the file. Please keep in mind that all the above listed properties may be modified by the programmer and the new settings will be acquired by the project when opened again afterwards (this means that you will need to close and reopen the project after having modified the ".Constraints" file).



A child project, when created for the same parent project's platform, inherits the same constraints as its parent project. However, if the child project is created for a different platform, ie. the parent project is Win32 type and child project is WinCE, the child project will inherit the constraints of its platform, WinCE.constraints, found in the Movicon installation folder.



When a new project is being created with the Wizard, Movicon will search for an existing ".Constraints" file with the name of the project to be created. The search for this file is carried out in the Movicon installation folder. When it is found it is then copied in the projects destination folder with the project name and ".Constraints" extension. The files originally created in the installation phase are:

**Win32.Constraints:** file containing the Win32 project constraints

**WinCE.Constraints:** file containing WinCE project constraints

**j2se.Constraints:** file containing j2se terminal project constraints

**j2me.Constraints:** file containing j2me mobile phone project constraints

## New Project Wizard customizing

One of the most powerful tools, that Movicon has made available for use, can be used to create new and customized "wizards" as new supervision project creation models. This potentiality opens the way to "auto-planning", where new Movicon projects can be created ready with presettable components (Variables, resources, objects).

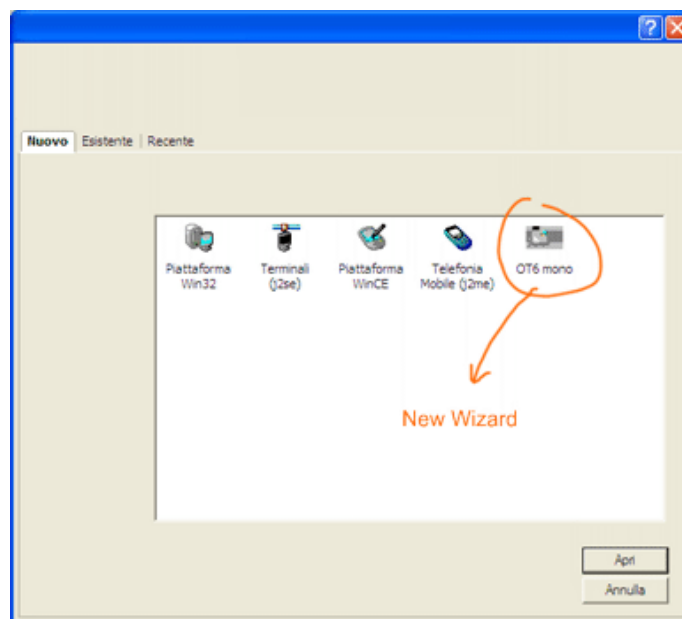
The wizard concept is based on the possibility to build a "macro" procedure, based on VBA codes, in order to create XML project files or any of its components.

By using this criteria, companies can preset the project's creation criteria, by creating for instance variables and their links, stiles and sizes of screens, their contents, the alarms, etc.

- **Once this criteria has been established and setup, therefore the VBA macro, you can click on the corresponding icon to get a complete creation of the project according to the set features and functionalities.**

When creating a "New Project, "Movicon will search for the 'wizard' sub-folder (which needs to be created), where the program is installed, for all eventual files with the ".movwiz" extension. Those present will be listed as additional icons in the Movicon "New Project" dialog window. If, apart from these files, exists another file with the same name and an .ico extension, it will be used as the icon to be displayed in the window otherwise Movicon will use a default icon.

The figure below illustrates the creation window of a new project. In addition to the standard Movicon projects, you will see a new icon corresponding to a custom project creating wizard.



The .movwiz file must be in **unicode text format** containing the basic code with the following entry function:

```
Sub OnWizard(bRet As Boolean, ByRef szFilePath As String)
```

```
End Sub
```

The bRet parameter when set at True will allow Movicon to proceed with opening the file passed with the szFilePath parameter.

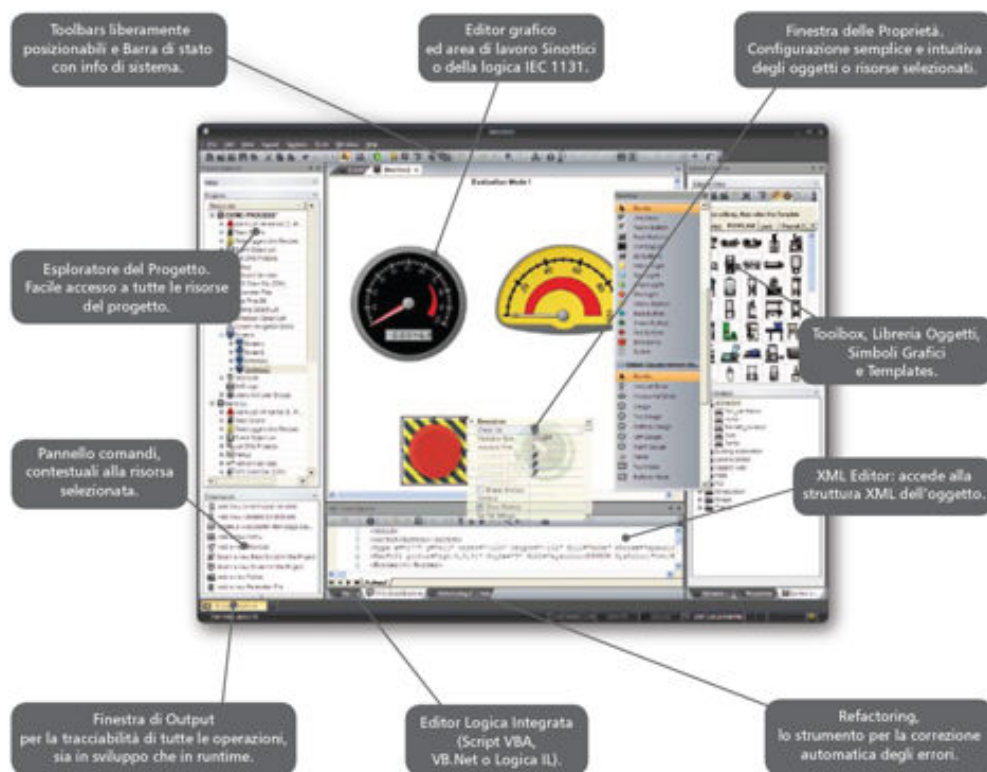
The necessary operations are to be done within this basic procedure (including the use of dll, ocx etc) to create the various projects files to be then opened by Movicon.

Usually, XML files creating is based on the Microsoft XML Parser.

Contact technical support to get further information and/or wizard examples.

## The Workspace

The workspace provided for the programmer is completely customizable. By using the commands from the 'View' menu, or the right mouse key, you can select the windows and the toolbars to be displayed for configuring the programming interface as you please.



The **Tools Bar**, **Menu** and the **Project Windows** can be positioned inside the workspace where you please. The Project Windows can be **"Docked"** in order to keep them in the workspace's foreground.

## MDI Tabs

Editing resources which open a window in the workspace, such as Screens, String Tables, Basic Scripts, etc., allow a series of MDI Tabs to be displayed at the top of the window to allow you to quickly pass from one resource to another:

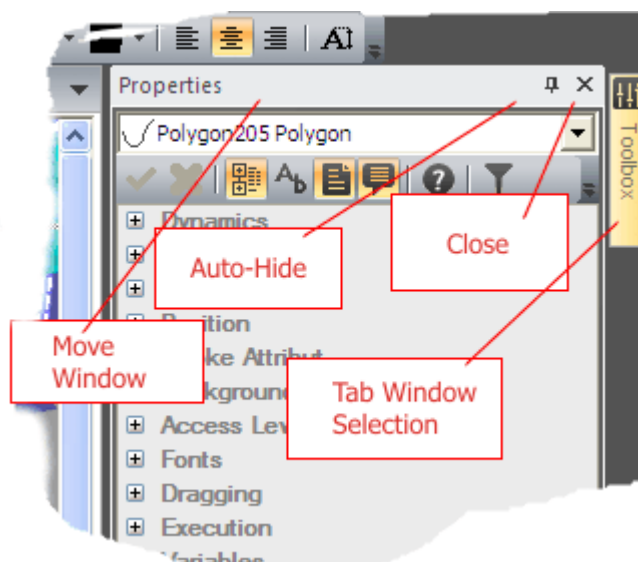


- Right clicking on these Tabs a context menu is open, containing the following commands:
  - **Close:** closes the resource in question
  - **Go to Resource:** highlights the resource in question in the "Project Explorer" tree. This command is useful when resources of different projects or child projects are opened at the same time, to identify the project the resource belongs to, especially if resources belonging to different projects have the same name.
    - **Resource TAB Groups**
  - **New Horizontal Tab Group:** opens a new tabs group and tiles the groups horizontally.
  - **New Vertical Tab Group:** opens a new tabs group and tiles the groups vertically.

## Docking Project Windows

The Project Windows are the fundamental tools through which the programmer can interface with the project. Each window has a precise function, such as gathering and displaying the project resources, setting or modifying the components' properties, editing script codes, etc.

As the Project Windows are sources of precious information and are therefore used frequently, they can be kept active to be displayed continuously, independently of the resource or screen you are working with. The windows which are kept constantly active can then be set with the '**Dock**' or '**Hide**' modes.



When a window is "**Docked**" it will always remain visible and will occupy space within the workspace. Some project windows, such as the Properties window, when parked,

dynamically change their contents to adapt to the resource or the component being focused on.

When the window is '**Hidden**' only the Tab identifying it is visible and is positioned on one of the sides of the Movicon window. When the mouse cursor is positioned on the Tab desired the window is displayed like a drop-down menu. When the window is no longer being focused on it will automatically be hidden again.

To turn the window from being '**Docked**' to '**Hidden**' and viceversa, simply use the '**Pin**' button on the top right title bar at the side of the "**X**" button which closes the window. The status of the button's icon shows which display mode is in action:



This icon means that the window is 'Docked', therefore it will remain in the foreground and always visible in the position defined by the programmer.



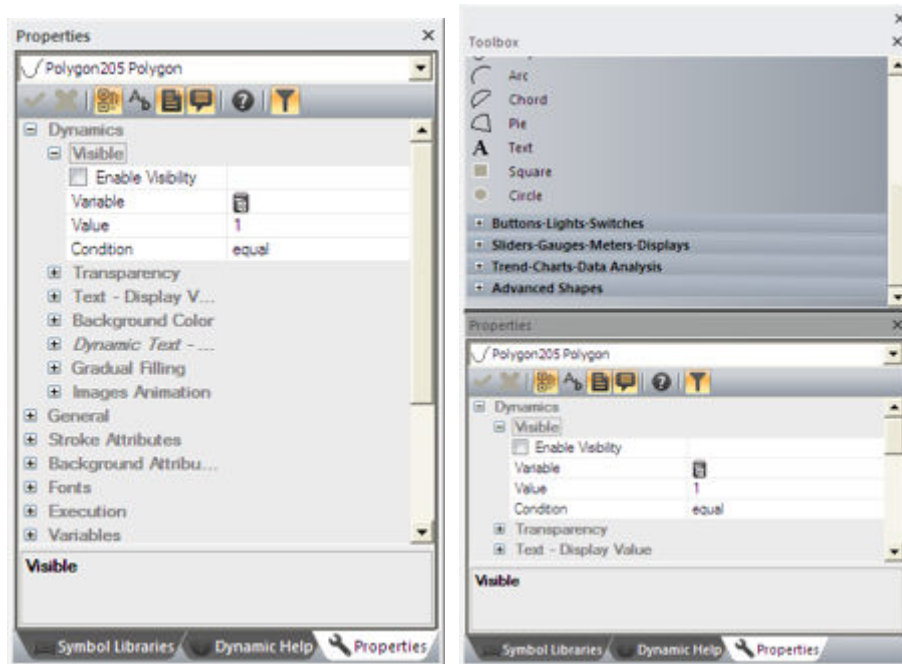
This icon means that the window is 'Hidden', therefore only the Tab with its name remains visible at the side of the screen. When the mouse cursor is positioned on the Tab, or clicked on, the relating window will be displayed in its entirety. When changing focus to another window or area in the workspace, the window will automatically be hidden again.

The Movicon Project Windows which are subject to these settings are:

- **Project Explorer:** this window contains all the project's resources
- **Properties Window:** this window is used for editing or modifying the properties of each project resource and component
- **Script explorer:** this window is used for editing the project's components' script codes
- **Logic Explorer:** this window is used for editing the project's components' codes in PLC Logic format
- **Symbols Library:** this window is used for accessing the Movicon Power Template symbols
- **Toolbox:** this window is used for accessing lists of drawings, controls and components which can be inserted within project Screens
- **Dynamic Help:** this window reports the main links to help topics inherent to the component being focused on
- **Output:** this is the window where all the system messages are reported by Movicon, such as errors, alerts etc., which are checked during programming or runtime
- **Watch:** this is the project debug window which is displayed during the Runtime phase. Variables can be forced or the project can be debugged through this window

When a window is docked you can drag it to a more suitable position either by attaching it to the Movicon window border or moving it to the centre of the Movicon workspace or to any other area. In order to do this drag the window by its title bar.

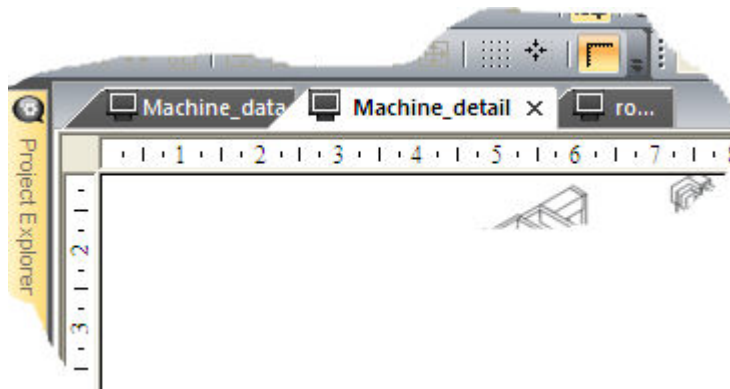
In addition to this you can also create compositions with more windows. By dragging one window onto another and dropping it into a suitable position you can get two composition types. In the first case you get one single window where the Tabs of the component windows are visible at the bottom and by clicking on the one desired will activate its relating window. Whereas in the second case, the windows are all visible but are attached to one another to form one window only:



Above are shown two configuration which can be obtained by overlapping two or more project windows. The programmer can choose the mode most suitable to his requirements in the workspace.

## The Rulers

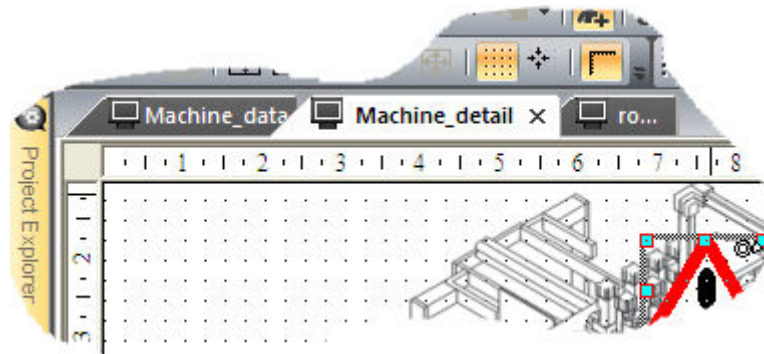
The Movicon Screen windows can be given lateral '**Rulers**' to facilitate designing the graphical interface.



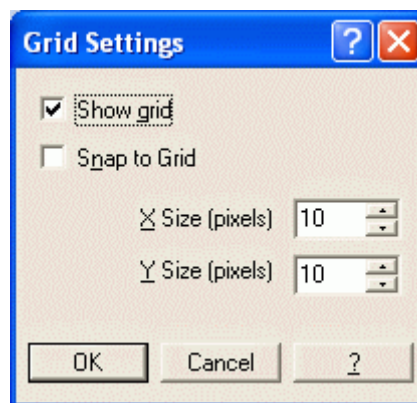
To enable or disable the Rulers just simply use the appropriate command found on the "**Aligning Bar**", or use the "**Ruler**" item from the "**Layout**" menu. When using the menu item you can enable the horizontal and vertical rulers separately as well as setting the referential measure units (Millimetres, Centimetres, Inches).

## The Screen Grid

The Movicon Screen windows can be given "**Grids**", to make designing the graphical interface and in particular any alignments easier. To enable or disable the Grid display, simply use the appropriate command found on the "**Aligning Bar**", or use the "**Grid Settings...**" item from the "**Layout**" menu.



In addition to the command for making the Grid visible or invisible there is also another command to enable the alignment of objects with the Grid. This command can also be found on the "**Aligning Bar**", or you can use the "**Grid Settings...**" item from the "**Layout**" menu. When using this menu you can also set the Grid's sizes, in pixels, being the space you wish to obtain between one point of the Grid and the next.



### Show Grid

Enables or disables the displaying of the grid on screen.

### Snap to Grid

When enabling this setting the screen's objects will be aligned to the grid according to the grid's resolution.

### Size X (pixels)

The distance in pixels between one point on the grid's horizontal axis and the next is set in this field. Obviously lower the value, more precise the resolution will be.

### Size Y (pixels)

The distance in pixels between one point on the grid's vertical axis and the next is set in this field.

Obviously lower the value, more precise the resolution will be.

## The Movicon Grid

Every screen has a second grid (different from the "The Screen Grid" which is customizable through the "Layout Menu-Grid Settings...") which is 1x1 in size and is always present but not visible. This grid is used by Movicon for forcing Snaps (alignment) of objects when the "Snap to Grid" property is enabled.

This second grid is handy in cases where using the zoom for positioning objects in areas that, with a normal 100% zoom, does not correspond to a pixel exactly. When using the zoom, therefore objects moved with the mouse or the keyboard direction arrow keys cannot be moved as pleased, but are moved according to the base multiple of pixels at 100% zoom. For instance when zoomed at 200%, each move will correspond to double the number of pixels with zoom at 100%.

The object's real sizes and position, being those corresponding to zoom = 100%, are always displayed in the Movicon Status Bar. When a multiplicative zoom (ie. 500%) is active, the values in the object's Properties Window (Position X, Position Y, Width, Height) are also multiplied by the zoom's factor. For instance, a rectangle sized 20X10 zoomed at 500% will have "Width = 100" and "Height = 50" in its Properties Window. If, while zooming, the values of these properties are changed using values that are not multiplied by the base unit (ie. 5 pixels), it will not snap back to grid correctly.

## Customizing Menus and Toolbars

---

Movicon allows you to customize the system menu and toolbar configurations to accord with your personal user styles.

By accessing the 'Customize' command from the 'Tools' menu, which opens by right mouse clicking in the workspace, the system settings window will display as illustrated in the figure below.

Each different setting card can be accessed through this window by clicking its corresponding Tab to activate it.

### Commands

You can view the complete list of menu commands, which are available in both in the programming and runtime phases, on the Commands Card.

These commands cannot be modified or their order changed.

### The Toolbars

From the Toolbar card you can view the complete list of the Movicon toolbars, which can be enabled or displayed by using the selection button found on the side.

You can restore the default positions of all the Toolbars by using the Reset All command.

New toolbars can be added but not commands.

### Tools

By using the Tool card you can customize the available commands in the Tool commands menu containing the utility applications which can be launched. The commands from the Tool menu can be customized, by adding new items or by modifying the existing ones. Each menu item has a command line ('Commands') with the relative argument, which will be executed when the command is activated from the menu.

By using the tools from the bar at the top of the card you can add or delete the item from the Tools menu. The list of applications existing on the menu are initially loaded from the "Tool.xml" configuration file (see paragraph on "Tools Menu Configuration").

### Keyboard

From the Keyboard card you can view the complete list of keyboard commands of the Movicon Toolbars.

The default settings can be restored by using the Reset All command.

The keyboard commands can be removed but not added to.

### Menu

You can customize the display of the system menu through the Menu card. The Project Menu selection is not active at this time.

The 'Application Frame Menus' command lets you defined the type of animation you wish to use each a menu is opened.

The 'Menu shadows' box lets you enable or disable the menu's shadow when opened.

### Options

The Options card lets you customize the further use of the menus in Movicon. Through the selection box, you can set the style in which you wish the menus to be displayed.

The 'Large Icons' box lets you enlarge the sizes of the icons from the Toolbars.



The 'Visualizations' box lets you enable the tooltips when the mouse is pauses on the icons of the toolbars.

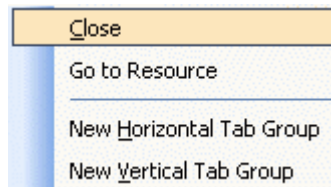
The 'Menus show recently used commands first' command lets you set how to manage the menu commands being used, by defining whether to immediately display only those used the most or those recently used.

## Resource TAB Groups

When having various resources opened at the same time during the project editing phase, a series of TABS will be displayed with which you can use to pass from one resource to another very quickly. This series of TABS is called the **"TAB Group"**.

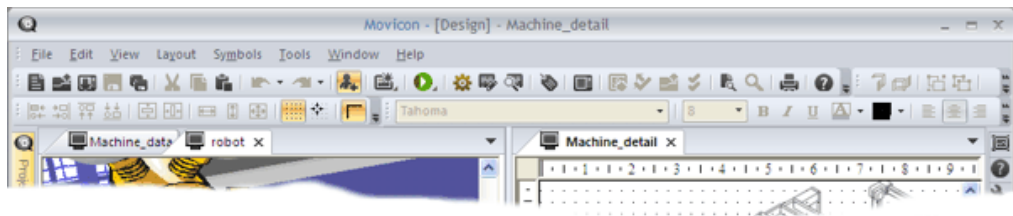


When right mouse clicking on the TAB group a menu will appear to let you execute certain commands:



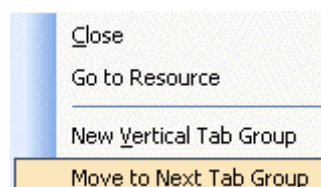
The **"Close"** command closes the selected resource, whereas the **"Go to Resource"** command highlights in the 'Project Explorer' window which resource is active in the workspace.

The **"New Horizontal TAB Group"** and **"New Vertical TAB Group"** commands are used for creating new horizontal or vertical TAB Groups so that more resources can be displayed at the same time and regrouped as pleased if required:



In the example above a second vertical TAB group has been created to group the screen and Basic Script resources separately.

Once two or more TAB groups have been created you can move a resource from one group to another by simply clicking and dragging it to the other group. This can also be done by using the **"Move to Next TAB Group"** command found on the menu which appears when clicking the right mouse button on the resource's TAB:





# Tool Bars

---

Movicon provides a few **Tool Bars** which can be displayed or hidden by using the items from the **"View Menu"** or by right mouse clicking on the tool bars in the area they occupy.

The functions which can be executed from anyone of the tool bars are described in the relevant sections. Below we will briefly describe the how the bars work according to the Windows standards. The tool bars, after being displayed, can be dragged to any point of the workspace, and can be sized as pleased.

Movicon also lets you customize the bars to display, for example, only a few of the commands by using the button found at the far right of the bar. The automatic bar reset, for reactivating all the commands for default, can be done with the **"Reset Toolbar"** command which is situated in the bar's properties or in the **"Customizing Menu and Toolbars"** window. To return the bar back to its original position just double-click the title of each single Toolbar or on the icon on the bar's far left.



Double-clicking this tool bar icon will position the bars onto the workspace where they can then be dragged and moved to any area of the screen desired.



Double-clicking on this icon will open a menu through which you can customize the bar by enabling or disabling the display of commands or reset any setting already carried out.

All the tool bars support the right mouse key, through which the same commands can be set from the **View** menu.

The available Tool Bars are:

**Tools Bar**  
**Aligning Bar**  
**Formatting Bar**  
**Symbols Bar**  
**Layer Bar**  
**Menu Bar**  
**Status Bar**

The Tool bars also show the commands on the Movicon Menus. Therefore the tool bars are used for getting to the main commands (those frequently used) directly with a press of a button to speed up project editing operations.

## Tool Bar

---

The Movicon Tool Bar reports the major part of the commands existing in the **"File Menu"** and in the **"Edit Menu"**.



### New (Ctrl+N)



Opens a new project by presenting the wizard window to create it.

This command is also available in the Movicon **"File Menu"** of Movicon.

### Open (Ctrl+A)



Opens an existing project, by presenting the browse window for you to search for it.

This command is also available in the Movicon **"File Menu"**.

### Open Device Project



Opens the Download window to consent downloading of project from the CE device connected.

This command is also available from the Movicon **"File Menu"**.

#### **Save (Ctrl+S)**



Saves the modifications made to the current project resource, which the one selected in the "Project Explorer" window at that moment.

You can save a project even when the license inserted, hardware or software, is only runtime type. In order to do this, you will need to keep the "D" key pressed down and then activate the project save command. This will save the project and enter the application into "Demo Mode". Close and reopen Movicon to exist from "Demo Mode".

This command is also available in the Movicon **"File Menu"**.

#### **Save all**



This makes a complete save of all the projects opened in the "Project Explorer" window.

This command is also available in the Movicon **"File Menu"**.

#### **Cut (Ctrl+X; Uppercase+Cancel)**



Cuts the object, resource or any other thing which has been selected by cancelling it and saving it in memory on the Windows Clipboard.

This command is also available from the Movicon **"Edit Menu"**.

#### **Copy (Ctrl+C; Ctrl+Ins)**



Copies the object, resource or any other thing which has been selected in memory on the Windows Clipboard.

This command is also available from the Movicon **"Edit Menu"**.

#### **Paste (Ctrl+V; Uppercase+Ins)**



Pastes the object, resource or any other thing which has previously been Copied or Cut on to the Windows Clipboard on the point selected by the mouse in the workspace.

This command is also available from the Movicon **"Edit Menu"**.

#### **Undo (Ctrl+Z; Alt+Backspace)**



Undoes the last operation executed, (ie. cancelling of a resource, editing of an object, etc.). By clicking on the arrow on the right hand side of the icon a window will appear showing the last operations executed in chronological order. More than one operation can be selected and cancelled at the same time.

This command is also available in the Movicon **"Edit Menu"**.

#### **Redo**



Reverses the last Undone operation (ie. the cancelling of a resource, editing of an object, etc). By clicking the arrow on the right hand side of the icon a window will open showing the last cancelled operations in chronological order. More than one operation can be selected and restored at the same time.

This command is also available in the Movicon **"Edit Menu"**.

#### **Disable User In Debug**









By using this button you can disable or enable the user management in the Debug, by putting the project in Runtime from the development environment. This function is very handy when carrying out project tests without having to activate or deactivate the password management. This also safeguards against accidentally leaving the password management disabled after having completed the test run.

#### **New Resource**





This command allows you to insert a new Multiple Resource in the **"Project Explorer"** window. In this case, five resources can be inserted with the possibility of creating simple folders. By keeping the mouse button pressed down on the icon, for at least a second, a

drop down menu will open containing six icons for selecting the resources to be inserted. Normally the tool bar displays the icon of the last resource selected. The choices are as follows:

-  **New Menu.** Inserts a new menu into the previously selected folder from '**Menu**' Group in the '**Project Explorer**' window.
-  **New Shortcut.** Inserts a new Shortcut into the previously selected folder from '**Shortcuts**' Group in the '**Project Explorer**' window.
-  **New Basic Script.** Inserts a new Basic Script into the previously selected folder from '**Basic Script**' Group in the '**Project Explorer**' window.
-  **New Screen.** Inserts a new Screen into the previously selected folder from '**Screens**' Group in the '**Project Explorer**' window.
-  **New Parameter File.** Inserts a new Parameter File in the pre-selected folder within the "**Parameter File**" in the '**Project Explorer**' window.
-  **New Folder.** Creates a new folder in the pre-selected position within the "**Multiple Resource**" in the '**Project Explorer**' window. Structuring "Multiple Resource" groups with folders and sub-folders helps you organize the project's resources better but does not change its way of functioning.


### Project Run

This command allows you to Run the project. There are four runtime modes that can be executed from the management environment. By keeping the mouse button pressed down on the icon for at least one second you will get a drop down menu containing the four icons for selecting the mode to be activated. Normally the last selected Run modality is displayed in the Tool Bar. The choices are the following:

-  **Start Project.** Executes the project Runtime in the usual way.
-  **Run Screen.** Only runs the project's graphics. This means that only the graphical and animation part of the project will be run and not the Communication Drivers, the Data Loggers, etc.
-  **Step Mode.** The project is run one step at a time. The programmer has to confirm the execution of the next step. The same goes for the project stop phase. This modality is very handy to identify which phase generates the error when problems arise at the project startup.
-  **Attach Running.** When using this mode, Movicon will propose a browse window from which you can select a PC, local or network, on which a Movicon project is already running. You will then be able to interact with the project by using the "**Watch Window**" for executing any debugs for instance.

This command is also available from the Movicon "**File Menu**".

### Edit RealTime DB Variable (Alt+D)

-  This opens the window listing the project's RTDB Variables.

This command can also be found in the Movicon "**View Menu**".

### Edit String Table (Alt+S)

-  This opens the "String Table" for editing the project strings

This command can also be found in the Movicon "**View Menu**".

### Check Missing Strings (Alt+M)

-  This activated the Inserting Missing Strings command for the selected resource.

This command can also be found in the Movicon "**View Menu**".

### Options Used



This opens the dialog window to check the options of the licence being used.

This command can also be found in the Movicon **"View Menu"**.

### Upload Project



This opens the dialog window for uploading the project to a CE device.

### Get Latest Version



This command allows you to read the last version of the resource selected in the Project's Explorer Window with the project in Visual SourceSafe.

This command can also be found in the Movicon **"File Menu"** and in the resource menus.

### Check Out



This command allows you to extract a resource file from the SourceSafe project.

This command can also be found in the Movicon **"File Menu"** and in the resource menus.

### Check In



This command allows you to put back a resource file in the SourceSafe project.

This command can also be found in the Movicon **"File Menu"** and in the resource menus.

### Undo Check Out



This command allows you to undo the changes made to a resource.

This command can also be found in the Movicon **"File Menu"** and in the resource menus.

### Find Next



After having started a search with the Find button you can carry out another search for the next specified text with the Find Next command, also executable with the F3 key.

This command is also available from the Movicon **"Edit Menu"**.

### Find



This is made available after a resource, set with this tool, has been opened. The 'Find What' command permits you to specify a character or text string to be searched for within the selected resource.

This command is also available from the Movicon **"Edit Menu"**.

### Print (Ctrl+P)



This button is used for printing the currently opened screen window. A window will appear to select and set the printer.

This command is also available from the Movicon **"File Menu"**.

### Help (F1)



This button activates the Movicon online help directly on the topic belonging to the description of the current resource, component, etc. being used.

## Aligning Bar

---

The Movicon **Aligning Bar** mostly shows the commands presented in the "**Layout Menu**". By using these commands you can align and side the diverse components and designs inserted in the screens.



Aligning objects on screen can also be done according to their baricenter. You can get hold of this function by keeping the "Shift" key pressed down and selecting the command from the "toolbar" or from the aligning menu.

### Align Left



This command aligns the selected objects to the left hand side of the screen.

### Align Right



This command aligns the selected objects to the right hand side of the screen.

### Align Top



This command aligns the selected objects at the top of the screen.

### Align Bottom



This command aligns the selected objects at the bottom of the screen.

### Centre Vertically



This command centres the selected objects vertically in the screen's area. When more than one object are selected, they will be centred in the area they occupy.

### Centre Horizontally



This command centres the selected objects horizontally in the screen's area. When more than one object are selected, they will be centred in the area they occupy.

### Same Width



This command resizes with the same width of the objects selected on screen.

### Same Height



This command resizes with the same height of the objects selected on screen.

### Same Size



This command resizes with the same width and height of the objects selected on screen.

### Grid ON-OFF



This command lets you activate or deactivate the Grid display on screen.

### Snap to Grid



This command lets you activate or deactivate the objects' alignment to the Grid on screen. This command also works when the Grid is not displayed, which means that the objects will be aligned according to the Grid's settings.

### Rulers



This command lets you activate or deactivate the display of the screen's Rulers.

## Symbol Bar

---

The Movicon **Symbol Bar** shows some commands that are found in the "**Symbols Menu**". By using these commands you can change the tab order of the different components and drawings inserted on screen and manage symbol creations.



### Move Next



This command is used for moving the selected object one place forward in the overlap order.

### Move Previous



This command used for moving the selected object one place back in the overlap order.

### Create Symbol



This command creates symbols. For further information, please refer to the section on "**Commands for Creating Symbols**".

### Ungroup



This command ungroups symbol formations. For further information, please refer to the section on "**Commands for Creating Symbols**".

### Re-Create Symbol



This command re-creates the symbol. For further information, please refer to the section on "**Commands for Creating Symbols**".

### Add Symbol to Library



This command adds the symbol at the Template Library. For further information, please refer to the section on "**Commands for Creating Symbols**".

### Dynamic Property Inspector



This command opens the "**Dynamic Property Inspector**" window relating to the selected symbol.

### Edit Image...



This command allows you to open the back image associated to the screen in edit mode.

## Layer Bar

---

The Movicon **Layer Bar** allows you to activate or deactivate the screen page Layers in programming mode. When the layer management is enabled you will be able to display or hide the objects associated to the layers.



### Layer



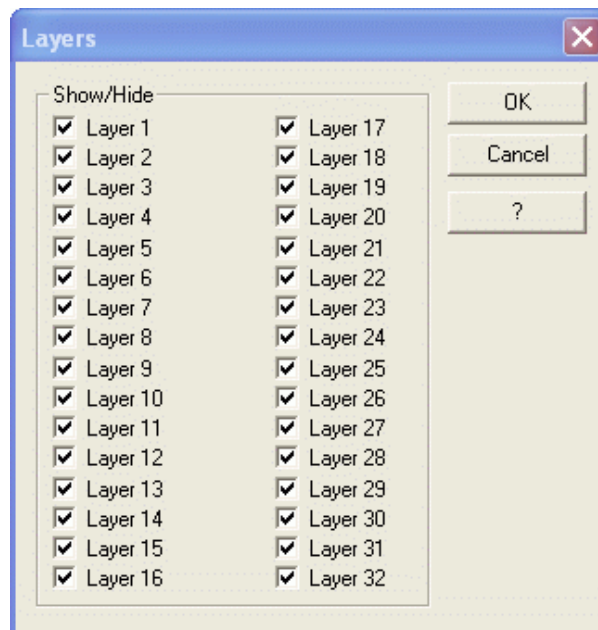
Each one of the buttons shown enable or disable the indicated level. Only the first 16 levels have been provided for space reasons. To access the other 16 layers you need to use the "**Set Layer Display Mode**" button.

### Set Layer Display Mode



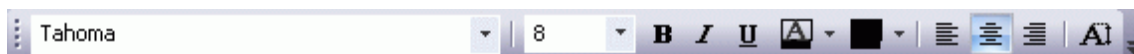
This command opens the dialog window which allows you to display/hide all of the 32 levels.

The "Set Layer Display Mode" button opens the following dialog window where you can set the visibility of all the screen's 32 layers.



## Formatting Bar

The Movicon **Formatting Bar** is needed for editing and configuring the texts in the controls and drawings inserted in the screens.

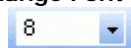


### Change Font



By using this drop down list you can select the Character Font to be set. The list shows the Operating System's Fonts.

### Change Font Size



By using this drop down list you can select the character size to be set. The list reports the values based on the Font selected in the 'Character Font' box.

### Bold



This command activates or deactivates the displayed text, from the selected component or symbol, in Bold.

### Italic



This command activates or deactivates the italic property of the displayed text from the selected component or symbol.

### Underline



This command activates or deactivates the underline property of the displayed text from the selected component or symbol.

### Text- Pen Color



This command is used for selecting the color for the displayed text from the selected component or symbol.

#### Back Color



This command is used for selecting a color to associate to the background of the selected component, symbol or screen.

#### Align Left



This command is used for aligning the text, displayed in the selected component or symbol, on the left. The alignment refers to the internal of the area occupied by the control or symbol in question.

#### Centre



This command is used for aligning the text, displayed in the selected component or symbol, in the centre. The alignment refers to the internal of the area occupied by the control or symbol in question.

#### Align Right



This command is used for aligning the text, displayed in the selected component or symbol, on the right. The alignment refers to the internal of the area occupied by the control or symbol in question.

#### AutoSize Text



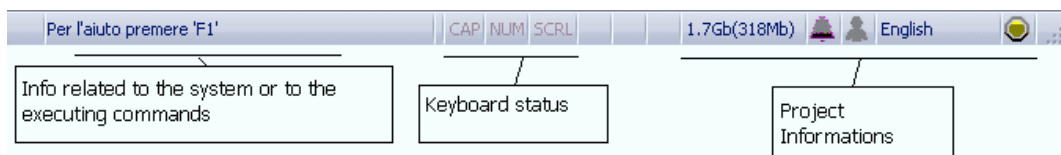
This command is used for activating or deactivating the AutoSize property to adjust the displayed text to the size of the selected component or symbol. When this function is enabled the 'Change Font Size' will no longer have effect, but the text will be adapted in proportion to the sizes of the component or symbol in order to cover the area made available.

## Status Bar

The **Status Bar**, found on the bottom border of the workspace, supplies the main information relating to the operating status of Movicon.

The status bar can be activated or deactivated by using the **Status Bar** command from the View menu.

The Status Bar will appear as shown below during the Project Runtime phase:



The icons presented on the right side of the bar indicate the Alarms Status, Users activation and the Communication Driver Status:



This field indicates how much virtual memory is available for the process. This value is updated every 5 seconds and is expressed in Kb, Mb or Gb according to the value. In Win32, this value depends mainly on the system paging file. However this value is more useful for monitoring in WinCE systems.



These two icons when shown blinking, indicate that there are active alarms or messages in the system. These icons will not show when there are no active alarms or messages.



This icon represents the log on status of users. When this icon is coloured it means that there is an active user in the system. Double-clicking this icon with the mouse will log-off the active user. When the icon is grey this means that there are no active users in the system. Double-clicking on the icon will open the authentication window to execute user log-on.



English

This text indicates which project language is active. By double-clicking on this icon, a dialog window will show where you can select the language to activate in the project.



This icon represents a 'Led' that when green/yellow means that the communication driver is working correctly, and when red means that the communication is not working correctly or has been interrupted. The Log will indicate the communication Driver problem type.

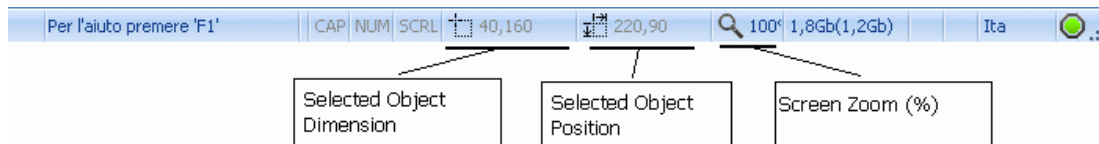
The text in the bar always indicates the operating status of Movicon or the description of the command being setup.

The operating status can be represented, for example, by the last alarm or message from the project.



The Status Bar can also display texts for command descriptions relating to the project's customized menus.

During the project design mode the Status Bar will show information on the positions of the objects selected:



When an object is selected on screen, its X and Y coordinates and sizes will appear on the Status Bar. Furthermore, by double clicking on the Zoom information (100% for default) you can change Screen's zoom percentage.

## RunTime Bar

The Movicon **RunTime Bar** is shown when the project is being run in the development environment only and allows you to execute certain debug commands and display the "Watch Window".



### Resume



When pressed this command resumes the run previously stopped with the Pause command.

### Pause



When pressed this command stops the following project processes:

- Project IL Logic
- IL Logic of Screens and the objects in them
- Event Objects
- Scheduler Objects
- Scaling Objects
- Data Loggers
- Alarm management
- Communication Drivers

Plus the status bar led starts blinking in yellow to indicate that the pause status is active.

This command does not effect Basic Scripts or OPC communications.

**Design Mode(ALT+F12)**



Aborts the project run and returns it to development mode.

**Toggle Local Project View (CTRL+F2)**



Shows or hides the project's local information display window in the "Watch Window".

**Toggle Statistic Project View (CTRL+F3)**



Shows or hides the display window for viewing the project's statistics in the "Watch Window".

**Toggle Watch Project View (tag)(CTRL+F1)**



Shows or hides the display window to view the debugging of the project's variables in the "Watch Window".

**Toggle Script Watch View**



Shows or hide the display window to view the project's basic script resources in the "Watch Window".

**Alarm Script Debugger...**



Opens the "Alarm Script Debugger..." window which links all the alarms managed in the project.

**Toggle Project IL Logic View (CTRL+F4)**



Shows or hides the display window to view the project's general IL Logic in the "Watch Window".

**Toggle local IL Logic View (CTRL+F5)**



Shows or hides the display window to view the project's Local IL Logic in the "Watch Window".

**Show or Hide the Soft Logic Debug window**



Shows or hides the SoftLogic debug window.

**Open Screen**



This command can be used to open any screen from the project in MDI mode.

**Run a Script**



This command can be used for executing any one of the project's Basic Scripts.

# Menus

---

Movicon has a **Main Menu Bar** which lets you access some of the Movicon principle commands. The Menu Bar is only available during the project development phase and is deactivated during Runtime. The Menu Bar can be dragged to any point in the workspace and can easily be returned to its original position by double-clicking the title bar or by dragging it again.



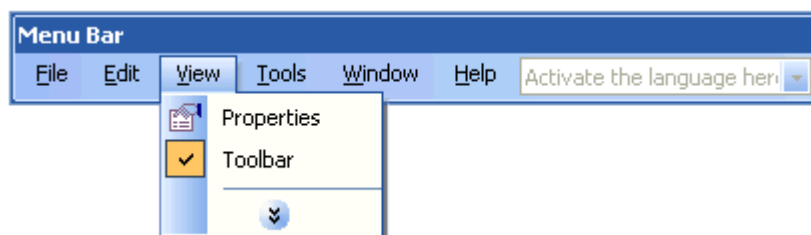
The sub-menus which are available from the Menu Bar are:

- File Menu**
- Edit Menu**
- View Menu**
- Layout Menu**
- Insert Menu**
- Symbols Menu**
- Tools Menu**
- Window Menu**
- Help Menu**
- Activate Language**

## Hidden Menus

---

Movicon, as consolidated in the Windows environment, hides the not so frequently used menu items to make the selecting of item lists much easier and intuitive.



To display all of the menu's items, you need to click the scroll arrows at the bottom of the item list to display the ones less used. To disable this functionality and display the whole list of menu items you have to access the settings of the **"Customize"** window.

## File Menu

---

The **File Menu** commands refer to the general project use. The menu items followed by dots indicate that the command, when selected, opens a system dialog window to let you set the options relating to the command. The name of the last project may also be shown in this menu (or the last 8 projects), to allow you to open them quicker. By selecting and activating the project name shown in the menu items (if present), determines the Open command of that same project.

### **New... (Ctrl+N)**

This executes the opening of a new project by presenting the wizard window to create it. This command is also available from the Movicon **"Tool Bar"**.

### **Open... (Ctrl+A)**

This executes the opening of an existing project, by presenting the browse window to search for the project desired.

This command is also available from the Movicon **"Tool Bar"**.

#### **Open device Project...**

This opens the Download window to allow project downloading from the connected CE device.

This command is also available in the Movicon **"Tool Bar"**.

#### **Close (Ctrl+A)**

Executes the closure of the resource being edited. If there are no resources opened, the active project will be closed instead.

#### **Set as Active Project**

When more than one project is opened in 'Project Explorer' window this command sets the one selected as Active Project.

#### **Start Project (ALT+F12)**

Project is put into Runtime mode.

This command is also available from the Movicon **"Tool Bar"**.

#### **Save (Ctrl+S)**

Executes the saving of any modifications done to the current project resources, being those selected in the "Project Explorer!" window.

This command is also available from the Movicon **"Tool Bar"**.

Projects can also be saved when the runtime license, hardware or software, has been inserted. In this case you will need to keep the "D" key pressed down and activate the project save command which will enter the application into "Demo" mode and save the project. To exit from "Demo" mode you will have to close and re-open Movicon.

#### **Save as... (Ctrl+S)**

A window will be displayed for renaming the project when executing a complete save of all the project. This command will save all the files contained in the DATA and IMAGES folder as well as the project's resources. The DATA folder, or the one set as the data folder, contained information on retentive variables statuses and Schedulers plans, which are usually handy to keep generally even after the project has been saved with another name. The IMAGES folder is also saved for obvious reasons. Folders created by the programmer for other reasons are not saved.

#### **Save All**

Executes the complete save of all the projects opened in the "Project Explorer" window.

This command is also available from the Movicon **"Tool Bar"**.

#### **Source Control**

By using this menu item you can access to commands for the project's SourceSafe management. For further information please refer to the relevant section.

#### **Print Setup...**

This command lets you set all the options relating to the type of printer used through an appropriate dialog window.

#### **Print Preview**

The Print Preview command lets you preview the active document page as it will be when printed. This command accesses to a display area in which other command buttons are made available and where the mouse pointer automatically turns into a zoom in the proximity of the print preview.

#### **Print (CTRL+P)**

The print command's job is to send the currently active resource, for example screen, to the printer. This command accesses a dialog window through which you can set the relevant print options. When you want to print a document different from the current one you must select it first.

This command is also available from the Movicon **"Tool Bar"**.

#### **Projects**

The last eight projects opened by Movicon are listed in this area of the window. You can directly click on the project name to open it without having to go to the selection window presented by the 'Open' command.

#### **Exit**

Executes exit from Movicon.

## Edit Menu

---

The Edit menu commands concern the use of some of the project resources and their availability depends on the function of the resource selected.

### **Undo (Ctrl+Z; Alt+Backspace)**

This cancels the last operation executed (eg. cancelling a resource, object modification, etc.). This command is also available from the Movicon **"Tool Bar"**.

### **Redo**

This command restored the last operation cancelled (i.e. cancelling a resource, object modification, etc.). This command is also available from the Movicon **"Tool Bar"**.

### **Cut (Caps+Canc; Ctrl+X)**

Cuts the object, the resource or anything selected. In this case the object is cancelled but stored in memory on the Windows Clipboard.

This command is also available from the Movicon **"Tool Bar"**.

### **Copy (Ctrl+C; Ctrl+Ins)**

Copies the object, the resource or anything that has been selected and stored in memory on the Windows Clipboard.

This command is also available from the Movicon **"Tool Bar"**.

### **Paste (Ctrl+V; Caps+Ins)**

Pastes the object, resource or anything that has previously been Copied or Cut to the Windows Clipboard on the point selected by the mouse within the workspace.

This command is also available from the Movicon **"Tool Bar"**.

### **Delete (Cancel)**

Deletes the select object without copying it in the Windows Clipboard.

### **Paste Special**

Inserts clipboard contents (paste) and a link to its source.

### **Find (ALT+F3)**

This is made available after a resource, set with this tool, has been opened. The 'Find What' command permits you to specify a character or text string to be searched for within the selected resource.

According to the standard procedures you can specify whether the text to be searched for must respect the lower/uppercase characters.

The 'Find Next' starts a new search, while the 'Cancel' button closes the search.

The Find command can also be accessed with the ALT+F3 keys from the keyboard, or from the **"Tool Bar"**, if available, through the Find editing box.

This command is also available from the Movicon **"Tool Bar"**.

### **Find Next (F3)**

After having started a search with the Find button you can carry out another search for the next specified text with the Find Next command, also executable with the F3 key.

This command is also available from the Movicon **"Tool Bar"**.

### **Replace**

The Replace command lets you specify a text to be searched for and a text to replace it when found.

### **Select All**

The Select All command lets you select all the resource's currently active contents at the same time when this is permitted.

### **Distribute Object Space...**

By using this command you can automatically position a series of objects on the screen so that they are equally distanced between each other by being organized in rows and columns.

For further information please refer to paragraph **"Distribute Object Space"**.

**Edit Text (F2)**

This command consents entry in edit mode of the selected object's Title.

**Import Draw...**

This command is used to import a vectorial drawing into the screen which has been realized with an external editor. Windows Metafile (EMF, WMF) is the format supported. For further information please refer to paragraph **"Importing/Exporting Vectorial drawings"**.

**Export Draw...**

This command is used to export a Movicon vectorial design in Windows metafile (EMF, WMF) format. For further information please refer to paragraph **"Importing/Exporting Vectorial drawings"**.

**Image Edit...**

This command opens the background image, associated to the selected screen or object in the workspace, in edit mode. This command opens the application predefined and associated to that file type (ie. "MS Paint" might be run if a ".bmp" file). This same command is also available from the menu which appears with a right click on the screen or on the object contained in the screen.

**Edit String Table...(Alt+S)**

Opens window for editing Movicon strings.

**Edit RealTime DB Variable ...(Alt+D)**

Opens window for displaying the Movicon List Variables.

**Check Missing Strings...(Alt+M)**

Executes the Inserting Missing Strings command for the selected resource.

**Check Options used (Dongle Requirements)...**

Opens window to check dongle options.

**Install this Project as Windows Service**

Installs Movicon as Windows Service so that when you start your PC Movicon will startup as Service.

**Uninstall from Windows Services**

Removes Movicon from the Windows Services. This command is only active when the project has been installed as Service beforehand.

**Compile Cross Reference**

Compiles the project's Cross Reference. This command is only available when the Cross Reference window is open.

## View Menu

---

The View Menu commands consent activation or deactivation of the main window displays and tool bars within the workspace.

**Full Screen**

This selection displays the Movicon workspace in full screen hiding the **Title bar**, the **Main Menu Bar** and the **Status Bar**. To restore the previous display just use the appropriate command key which appears in the workspace.



### Auto Load State Mode

This command allows you to enable/disable this behaviour mode for the Movicon editor windows. When this mode is enabled, the working environment status is loaded and saved in automatic according to the type of resources active. For instance, opening a screen will automatically activate the "Toolbox" window, display the "Symbol Bar" and hide all the other windows in the working environment. Predefined behaviour modes can however be customized by setting the working environment to another mode and then this new status will be saved and re-proposed when activating that type of resource. The resources which are loaded/saved with the working environment status are:

- No resource opened in edit mode
- Screen Window
- Basic Script code edit window
- Parameter File edit window
- Project variable edit window
- Project String edit window
- SoftLogic configuration window



Opening a resource while keeping the "Ctrl" key pressed restores the default windows configuration.

### Project Explorer

This selection displays the "**Project Explorer**" window. This command only works when the window is not currently displayed.

### Properties

This selection displays the "**Properties Window**". This command only works when the window is not currently displayed.

### Symbols Library

This selection displays the "**Symbols Library**" window. This command only works when the window is not currently displayed.

### Dynamic Help

This selection displays the "**Dynamic Help**" window. This command only works when the window is not currently displayed.

### Formatting

This selection displays or hides the "**Formatting Bar**" tool bar.

### Aligning

This selection displays or hides the "**Aligning Bar**" tool bar.

### Symbols

This selection displays or hides the "**Symbols Bar**" tool bar.

### Layer

This selection displays or hides the "**Layer Bar**" tool bar.

### Script Explorer

This selection displays the "Script Explorer" window. This command only works when the window is not currently displayed.

### Logic Explorer

This selection displays the "**Logic Explorer**" window. This command only works when the window is not currently displayed.

### Output

This selection displays the "**Output**" window. This command only works when the window is not currently displayed.

### ToolBox

This selection displays the "**Toolbox**" window. This command only works when the window is not currently displayed.

### ToolBar

This selection displays or hides the "**Tool Bar**".

**Status Bar**

This selection displays or hides the "**Status Bar**".

**Application Look**

This command allows you to select different ways on how you want the Movicon development environment to look by changing the windows' graphics and colors.

**Edit String Table...**

Opens the window for editing the Movicon Strings.

**Edit Real Time DB Variables...**

Opens the window for displaying the Movicon Variables List.

**Check missing Strings...**

Activates the Inserting Missing Strings command for the selected resource.

**Only Dynamic Object Properties**

This setting allows only animated objects to be displayed inside a selection of more than one component. This is needed particularly for being able to see only the animated objects in grouped symbols. Only the animated objects will be available from the Combo-box of the "Properties Window".

## Insert Menu

---

You will find commands for inserting strings and variables in the **Insert Menu**. This Menu is only available for some resources such as the "String Table", the "Variable List" window and the "Basic Script" window.

**New String (Ins)**

This command inserts a new string in the table. The string's ID is inserted with a progressive number with the following syntax: "String00001".

**New Language Column... (ALT+Ins)**

This command inserts a new column representing a new language. When the command is executed a input box opens for inserting the column's name (normally the name of the language in question).

**New Variable...**

Inserts a new variable in the Real Time DB.

**New Variable Group**

Inserts a new Variable group within the Real Time DB.

**New Structure Definition**

Inserts a new Structure Prototype in the Real Time DB.

**New Communication Driver**

Inserts a new Communication Driver in the Real Time DB.

**Add System Variables**

Inserts the Structure Prototype of the relative "**\_SysVar\_**" variable within the Real Time DB.

## Layout Menu

---

The Layout Menu commands refer to the use and management of the vectorial graphics contained in the screen and to the relative use of the window.



The layout of on screen objects can also be done according to their barycenter. This function is obtained by keeping the "Shift" key pressed down while selecting the command from the toolbar or from the layout menu.



**Alignment Objects**

This command opens a window for setting the objects' alignments.  
For further information please refer to the section on Movicon **"Object Alignment Functions"**.

**Center in Window**

This command opens a window for centering the objects.  
For further information please refer to the section on Movicon **"Object Alignment Functions"**.

**Set Same**

This command opens a window for setting the objects' sizes.  
For further information please refer to the section on Movicon **"Object Alignment Functions"**.

**Grid Settings...**

This command opens a window for setting the screens with the Grids.  
For further information please refer to the section on Movicon **"The Grid"**.

**Ruler**

This command opens a submenu for setting the screens with Rulers.  
For further information please refer to the section on Movicon **"The Rulers"**.

**TAB Order... (CTRL+D)**

This command lets you activated the Tabulation order of the objects on the screen.  
For further information please refer to the section on Movicon **"Tab Order"**.

**Resize**

Not available at this moment

**Remote Device Size (WinCE)**

This command lets you size the screen window with the target WinCE PC characteristics to which the project will be exported. Before launching this function you need to put into effect the connection between the desktop PC and the target PC with ActiveSync.

**Zoom In (CTRL+Più(Tn))**

This command enlarges the screen graphic.

**Zoom Out (CTRL+- (Tn))**

This command reduces the screen graphic.

**Zoom To**

The Zoom To function lets you enlarge a selected part of the screen.  
To select the part to be enlarged, mouse click on the first corner of the square and drag it.  
When releasing the mouse the screen will show an enlargement of the selected square.

**Refresh**

The Refresh command regenerates the video drawing.

## Symbols Menu

---

The Drawings Menu commands are used for changing or setting objects presented on the screen.

**First**

This command is used for placing the selected object in foreground.  
For further information please refer to the paragraph on **"Overlap Order"**.

**Last**

This command is used for placing the selected object in background.  
For further information please refer to the paragraph on **"Overlap Order"**.

**Move Next (+ (Tn))**

This command is used to move the selected object one place ahead in the overlapping order.

For further information please refer to the paragraph on **"Overlap Order"**.

### **Move Previous (-(Tn))**

This command is used to move the selected object one place back in the overlapping order.

For further information please refer to the paragraph on **"Overlap Order"**.

### **Symbol**

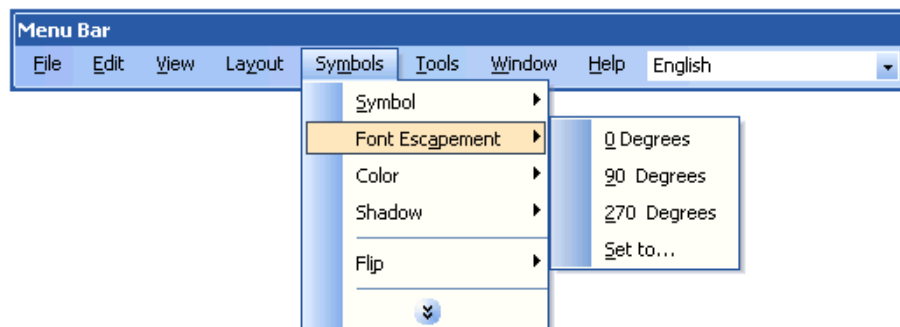
This command is used to group, ungroup or insert Movicon symbols in the library.

For further information please refer to the paragraph on **"Commands for Creating Symbols"**.

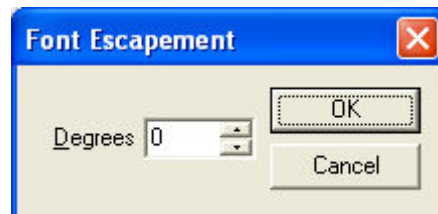
### **Font Escapement**

You can type in the text for the title of each object by using 'Properties Window'. The text will be made visible in the object itself. This property lets you establish the modifications for the font's standard direction, being zero degrees (Horizontal towards right).

The title font can be then set at 90° (vertical upwards), 270° (vertical downwards) or on a customized angle.



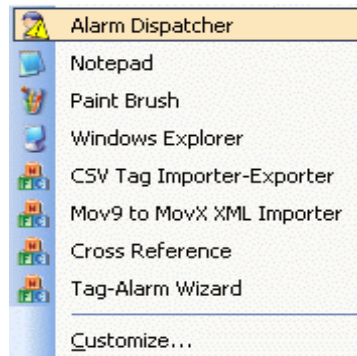
A customized angle can be inserted by using the appropriate dialog window as shown below:



### **Color**

The objects inserted on screen can be associated with the Colours desired which relate to the background, line around the edge or the text. Apart from using the 'Properties Window', the color settings can also be done by using the appropriate commands from the Drawing Menu or with the right mouse key. This command offers the advantage of speeding up the most common and frequent editing operations in drawings.

The Color is settable when the colour display has been enabled in the object's property otherwise the object will result transparent.

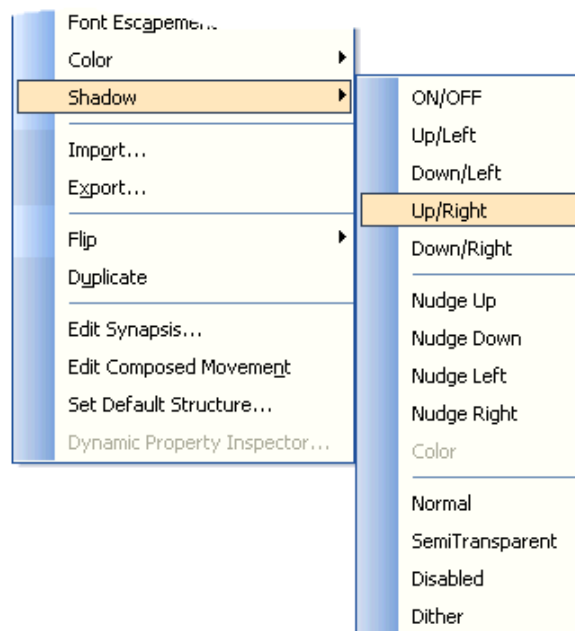


### Shadow

The objects inserted on screen can be associated with a Shadow effect by means of using the appropriate command from the Drawing menu or with the right mouse key.

The Shadow effect is settable when the shadow display has been enabled in the object's property.

The shadow will be displayed with the standard configuration and colour.



### Import...

This command is used for importing a vectorial drawing realized with an external editor onto the screen. The Windows metafile (EMF, WMF) is the supported format.

For further information please refer to the paragraph on **"Importing/Exporting Vectorial drawings"**.

### Export...

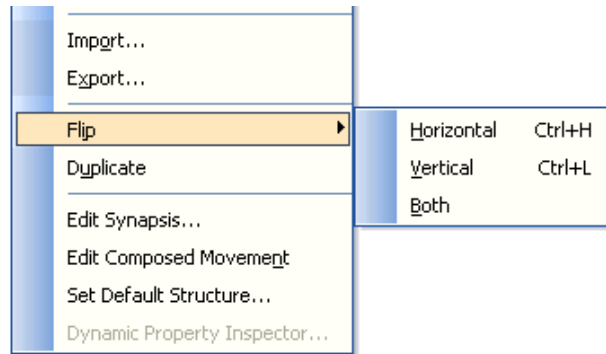
This command is used to export a Movicon vectorial drawing in Windows metafile ((EMF, WMF) format.

For further information please refer to the paragraph on **"Importing/Exporting Vectorial drawings"**.

### Flip

The Reflects command is used for turning over the selected drawing in Horizontal, Vertical or both mode.

These commands are settable from the Drawing Menu or by using the right mouse key.



### **Duplicate**

The Duplicate command is used for copying the object or the objects selected. Practically, this command is a 'copy-cat' of the system's Copy and Paste operations.

### **Edit Image**

This command consents you open the screen's associated back image in edit mode.

### **Edit Synapses...**

This command lets you create a new Synapses object. According to the block function editing techniques, the symbol's element can become an active working logic, linked in sequence to other Synapses function blocks.

For further information please refer to the paragraph on "Synapses".

### **Edit Composed Movement**

This command lets you edit the object's 'Composed Movement'.

For further information please refer to the paragraph on "**Composed Movement Editing**".

### **Set Default Structure...**

This command lets you associate a Default Structure to the selected object.

For further information please refer to the paragraph on "**Default Structures in symbols**".

### **Dynamic Property Inspector...**

This command lets you open the selected object's "**Dynamic Property Inspector Window**" window.

### **Compile IL Logic**

This command compiles the IL Logic for the selected symbol.

## **Tools Menu**

---

The **Tools Menu** shows the commands for executing frequently used applications. The list here below is the default list, but it can be customized editing the "Tools.xml" file (see "Tools Menu configuration") or using the "Customize" command (see "Customizing Tools Menus and Toolbars").

### **Upload/Download service**

Starts TCPUploadServer.exe.

### **Alarm Dispatcher**

Open the Alarm Dispatcher application session.

### **Notepad**

Opens a Notepad application session.

### **Paint Brush**

Opens a Paint Brush application session.

**Windows Explorer**

Opens a Windows Explorer application session.

**CSV Tag Importer-Exporter**

Opens a CSV Tag Importer-Exporter application session.

**Tag-Alarm Wizard**

Opens a Tag-Alarm Wizard application session.

**Development Code Info**

Runs the "RegDevCode.exe" application to read the Movicon development (or registration) code.

**Customize**

Opens the Movicon "**Customize**" setting window.

## Window Menu

---

The Window Menu commands refer to the usage and management of the Movicon resource windows or icons within the workspace.

**Window Split**

This command refers to the order of the windows present in the Movicon workspace. This command accesses to a selection Menu where you have to select either Vertical alignment or Horizontal alignment.

The Horizontal item puts all the windows, presented in the workspace, into order by aligning them horizontally across the screen starting from the left border towards the right border.

The Vertical item puts all the windows, presented in the workspace, into order by aligning them vertically down the screen starting from the top border towards the bottom border.

**Close All Documents**

The Close All command closes all the resource windows, currently active in the project's workspace, all at the same time.

**Workspace**

This command allows you to save the Movicon Workspace settings. In this way many configurations can be saved so that they can be called when needed. It is therefore possible, for example, to save one configuration where the project window have been docked in a certain way and another where the project windows have been set with hidden. By using this command you will be able to pass from one configuration to another at an instant. The commands available for menu item are:

- **Save:** save current configuration. You can specify the name to be assigned to the configuration or select an already existing one to overwrite
- **Load:** loads the selected configuration

**Window Name**

The last nine windows open the workspace are listed in this area of the window. When clicking on the name of the window with the mouse will bring it to the forefront and display it in full view.

**Window...**

This command is used for opening a dialog window containing a list of the window open in the workspace. This command is handy when more than nine windows are open in the workspace and therefore not all listed in the Menu.

The commands available in the this window are:

- **Activate:** activates the window selected in the list by bringing it into forefront vision in the workspace
- **OK:** closes the dialog window without activating any commands
- **Save:**
- **Close Window:** closes the window selected in the list

## Help (?) Menu

The Help menu indicated by the "?" character is the last item of commands available from the Menu bar.

By using Help the user can access system information and the on-line guide if available.

### Help Topics

This command opens the general Help window of the 'Movicon Online Guide'.

### Tutorial

Opens the Movicon tutorial.

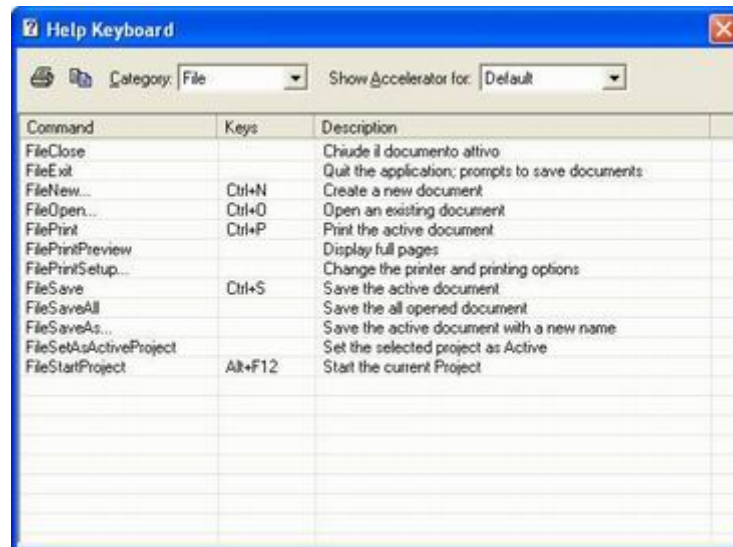
### Dynamic Help

This command opens the Movicon **"Dynamic Help"** window.

### Keyboard Map...

This command opens the window where the accelerators set for Menu commands are reported.

This window is only for consultation and not for making changes.



Command	Keys	Description
FileClose		Chiude il documento attivo
FileExit		Quit the application; prompts to save documents
FileNew...	Ctrl+N	Create a new document
FileOpen...	Ctrl+O	Open an existing document
FilePrint	Ctrl+P	Print the active document
FilePrintPreview		Display full pages
FilePrintSetup...		Change the printer and printing options
FileSave	Ctrl+S	Save the active document
FileSaveAll		Save the all opened document
FileSaveAs...		Save the active document with a new name
FileSetActiveProject		Set the selected project as Active
FileStartProject	Alt+F12	Start the current Project

### About Movicon...

By using this "Information on...", you can access information of the Movicon system being used which is very helpful when having to ask for any **"Technical Support"**.

The Progea Web and Email addresses are also available from this Information window, so that the user can get hold of them easily when needed or for requiring information on the most recent Movicon versions or its new products.

The **"Check for Updates"** button lets you check and download any product updates.

When selecting the **"Version Infos"** tab you can check all the information concerning the **version** of the project being used. As mentioned above, this information is fundamental to the **"Technical Support"** when contacting them for assistance.

### Online Forum

Opens a connection to the Progea Web Forum on Internet. In order for this to work the WEB connection must be active.

### Online Support

Opens a connection to the Progea Support Center Web page on internet. In order for this to work the active WEB connection must be active.

## Context Resource Menus

---

When right mouse clicking on one of the different resources available from the **"Project Explorer Window"** window, Movicon will provide you with a set of contextual Menus through which you can execute a certain number of commands inherent to the resource selected. All these command are also available from the Project Explorer's **"Commands"** window.

All the available commands are described below.

### **Open Project...**

Executes the opening of an existing project by presenting the browse window to search for the one desired.

This command is also available from the Movicon **"Tool Bar"**.

### **New Project...**

Executes the opening of a new project by presenting the wizard window to create it.

This command is also available from the Movicon **"Tool Bar"**.

### **Close (Ctrl+A)**

Closes the opened resources and the active project.

### **New Child Project...**

Executes the insertion of a child project by presenting the wizard window to search for and create the project desired. In this case you can insert an existing project or create a new one.

### **Remove Child Project**

Eliminates the selected child project. This operation simply removes the child project from the list of child projects, but it remains integral and is not cancelled.

### **New Variable...**

Inserts a new tag in the Real Time DB. Before insert the variable Movicon allow to edit the variable name. The default name is "VAR" followed by a progressive number "00001" (i.e. VAR00001).

### **New Variable Group...**

This inserts a new Variable Group within the Real Time DB. The purpose of having Variable Groups is only for organizing variables together. Variables which are linked to the same device, for example, are grouped together in folders. Before inserting a Group of variables, Movicon will display a dialog window where a name can be assigned to the Group. The "GROUP" name is proposed for default followed by a progressive number "00001" (eg. GROUP00001).

### **New Communication Driver...**

Inserts a new Communication Driver in the Real Time DB. The command opens a dialog window to select the Driver from those available.

### **Install this Project as Windows Service**

Installs Movicon as Windows service. In this way Movicon will startup as Service when the PC is started.

### **Uninstall from Windows Services**

Removes Movicon from Windows service. This command is only active when the project is installed as Service beforehand.

### **Edit String Table...**

Opens the window for editing the Movicon Strings.

### **Check Options Used (Dongle Requirements)...**

Opens the window for checking the options contained in the License.

### **New Structure Prototype**

Inserts a new Structure Prototype internal the Real Time DB. The Prototype will be inserted with the 'STRUCT' name followed by a progressive number "00001" (i.e. STRUCT00001).

**Add System Variable**

Inserts the Structure Prototype and the relating **"\_SysVar\_"** variable internal the Real Time DB,. This particular variable is managed by Movicon and reports a series of system information which may be very useful for the programmer.

**Compile Cross Reference**

Activates the project's cross reference compilation and automatically opens the window displaying the Cross Reference for variables.

**Associate a Data Logger/Recipe to this Variable**

Executes the association of a Data Logger or Recipe to the variable selected.  
For further information please refer to the section on **"Associating Data Loggers and Recipes to Variables"**.

**Associate an Alarm to this Variable**

Executes the association of an Alarm to the variable selected.  
For further information please refer to the section on **"Associating Alarms to Variables"**.

**Associate an Event to this Variable**

Executes the associated of an Event to the variable selected.  
For further information please refer to the section on **"Associating Events to Variables"**.

**Comm. Driver Settings...**

Opens a window for setting the parameters of the selected Communication Driver.

**Import Device Database...**

Opens the window for setting the symbolic file of the device (PLC, etc.).

**Add new Struct Member**

Inserts a new member variable in the selected Structure. The variable will be inserted with the **"VAR"** name followed by a progressive number **"00001"** (Eg. VAR00001).

**Change Struct Member Order...**

Opens the window for changing the order of the members internal the structure.  
For further information please refer to the section on **"Variable Member Order"**.

**Add a New Menu**

Inserts a new **"Menu"** resource internal the **"Menu"** group.

**Add a New Shortcut**

Inserts a new **"Shortcut"** resource internal the **"Shortcuts"** group.

**Add a New Script**

Inserts a new **"Script"** resource internal the **"Basic Script"** group.

**Add a New Screen**

Inserts a new **"Screen"** resource internal the **"Screens"** group.

**Add a New Folder**

Inserts a new **"Folder"** internal the Multiple Resources group.

**Add a New Parameter File**

Inserts a new **"Parameter File"** in the **"Parameter Files"** group.

**Add New Local Variable...**

This command allows you to insert a new local variable in the selected Screen.

**New Variable Script Event...**

This commands permits an event generated by a variable change to be inserted internal the Script code of the selected object or resource. Executing this command in fact opens a variable browse window enabling you to select the variable desired, after which Movicon will add an event defined as **"OnNameVariableChanged"** (Eg. OnVAR00001changed). This event will be called every time the variable changes its value during Runtime.  
For further information please refer to the section on **"Associating Variable Script Events to Screens"**, **"Associating Variable Script Events to Symbol"**.



**Create a WebClientX html page based on this Screen**

This command generates a HTML page which will be saved in the "\RESOURCE\ProjectName\ScreenName.html" Project Folder. The html page can then be used for linking up to the Server project as Web Client.

**Add New Shortcut Command**

Inserts a new Shortcut Command in the selected Shortcut resource.  
For further information please refer to the section on "**Shortcut Resource**".

**New Menu Item**

Inserts a new Menu Item in the selected Menu Resource.  
For further information please refer to the section on "**Movicon Menu Items**".

**New Popup Item**

Inserts a new Popup Item in the selected Menu resource.  
For further information please refer to the section on "**Movicon Menu Items**".

**New Separator Item**

Inserts a new Separator Item in the selected Menu resource.  
For further information please refer to the section on "**Movicon Menu Items**".

**Change Menu Item Order...**

Executes the opening of a dialog window for modifying the Item order.  
For further information please refer to the section on "**Movicon Menu Items**".

**Test Menu**

Executes a Menu Test during the programming phase.  
For further information please refer to the section on "**Test Menus**".

**New User Group**

Inserts a new User Group in the 'Users and User Group' resource.  
For further information please refer to the section on "**Users and Passwords management**".

**New User**

Inserts a new Users in the selected User Group.  
For further information please refer to the section on "**Users and Passwords management**".

**Add Users to Windows Local Domain**

Inserts users from the selected group in the PC's Local Windows Domain.

**Add a New Data Logger**

Inserts a new Data Logger in the "Data Logger and Recipe" resource.  
For further information please refer to the section on "**Data Loggers and Recipes**".

**Add a New Recipe**

Inserts a new Recipe in the "Data Logger and Recipe" resource.  
For further information please refer to the section on "**Data Loggers and Recipes**".

**Add a New Column**

Inserts a new column in the selected Data Logger or Recipe resource.  
For further information please refer to the section on "**Data Loggers and Recipes**".

**Import Database...**

Creates a Data Logger or a Recipe by importing the structure from an already existing Database.

**Create Recipe Editor**

Creates a Screen by automatically inserting the components (display, buttons, etc.) needed for displaying and changing the selected recipe. This command is very handy as there is no need for the programmer to create any recipe graphics.

**Add a new Alarm**

Inserts a new alarm object in the Movicon 'Alarm List' resource.  
For further information please refer to the section on "**Alarms**".

**Add a new Alarm Threshold**

Inserts a new threshold event for the selected alarm object.  
For further information please refer to the section on "**Alarms**".

**Add a new Alarm Area...**

This inserts a new Alarm Area. Before inserting the Area, Movicon will display a dialog window where you can assign the name of the Area. The "AREA" name is proposed for default followed by a progressive number "00001" (eg. AREA00001).  
For further information please refer to the section on "**Alarms Area**".

**Add a new Event Object**

Inserts a new Event Object in the Movicon 'Event Objects List' resource.  
For further information please refer to the section on "**Commands on Event**".

**Add a new Scaling Object**

Inserts a new Scaling Object in the Movicon 'Scaling Objects List' resource.  
For further information please refer to the section on "**Variable Scaling**".

**Add a new Scheduler Object**

Inserts a new Scheduler Object in the Movicon 'Scheduler Objects List' resource.  
For further information please refer to the section on "**Command Scheduler**".

**Add new RAS Station**

Inserts a new RAS Station.  
For further information please refer to the section on "**RAS Stations**".

**Add new Client Rules**

Inserts new Client Rules.  
For further information please refer to the section on "**Client Profiles**".

**Edit DCOM Settings**

Opens the Windows' window for the system's DCOM settings.

## Tools Menu Configuration

---

The **Tools Menu** can be configured by the programmer. There is a "Tools.xml" file in the Movicon installation folder which allows the Tool menu list to be customized. These tools can be Movicon add-ons and simple external executables.

The list of tools available in the menu is managed by using a Windows registry key. When Movicon is started up, it controls whether the tool list is already present in the Windows registry keys or not. If these keys have not yet been configured, Movicon will read the "Tools.xml" file and then compile the Windows registry keys with the attributes defined in this file. After this has been done, this file will no longer get read unless to remove registry keys. When Movicon is started up with the "CTRL" key pressed down, the registry keys will reset and the "Tools.xml" file will be read again.



After having modified the "Tools.xml" file you will have to restart Movicon keeping the "CTRL" key pressed down to activate them.

However, menu items can be added to the Tools menu using the menu's "Customize" command. In this case, modifications for adding or deleting applications from the menu are done directly in the Windows registry keys. This means that any resets to these keys due to restarting Movicon with the "CTRL" key pressed down, may cause the "Tools.xml" file to lose settings done using the "Customize" command when loading.

The "Tools.xml" file must be in this structure:

```
<?xml version="1.0" encoding="utf-8"?>
<ToolList>
  <Tool Label="Servizio Upload/Download" Arguments=""
  InitialDirectory="" Command="TCPUUploadServer.exe" />
  <Tool Label="Alarm Dispatcher" Arguments="" InitialDirectory=""
  Command="Dispatcher.exe" />
  <Tool Label="Tag Alarms Generator" Arguments="" InitialDirectory=""
  Command="AlarmGen.exe" />
```

```

<Tool    Label="Csv    Tag    Importer-Exporter"    Arguments=""
InitialDirectory="" Command="CSVTagIE.exe" />
<Tool    Label="Notepad"    Arguments=""    InitialDirectory=""
Command="Notepad.exe" />
<Tool    Label="Paint"    Arguments=""    InitialDirectory=""
Command="MsPaint.exe" />
<Tool    Label="Windows Explorer"    Arguments=""    InitialDirectory=""
Command="Explorer.exe" />
</ToolList>

```

where the attributes have these following meanings:

**Label** = name of application to appear on menu

**Arguments** = parameters to pass to the application in the command line

**InitialDirectory** = initial folder in which to execute the application search

**Command** = name of the application's executable file

When invoking a command from the Tools menu when the "InitialDirectory" attribute has not been specified, Movicon will set the current Windows folder (even if this installation path has been deleted from the Windows "PATH" environment variable). This behaviour consents will always execute the tool corresponding to Movicon, event when other products or add-ons, which install tools with the same name, are present. In cases where the tool to be executed is not present in the Movicon installation folder, it will be searched for in the folder defined in the Windows "PATH" environment variable folder.

# Project Explorer Window

All the Project's Resources are grouped in the '**Project Explorer**' window. This is the main window of the Movicon Workspace. This window allows you to display all the information relating to the resources which are the essence of the project itself.

The 'Project Explorer' window gathers all the Resource Groups into a tree structure. When selecting each single Resource Group or any of their sub-items, the properties of the object in question will be activated in the "Properties Window", through which you will be able to carry out any changes or necessary settings.

The "**Project Explorer**" can be activated with the 'Project Explorer' command from the 'View' menu.



You can use the "+" key from the number keypad for expanding or collapse a node from the project's tree structure list, displayed in the Project Explorer window.

This key functions independently with or without the Numlock, but the standard keyboard "+" key does not have this function.

When the selected node shows with a + sign next to it, pressing the "+" key will expand it. When the selected node shows with a - sign next to it, pressing the + key will collapse it back to its original state.

When the selected node has sub-nodes, the + sign will not display at its side.

The 'Project Explorer', as all the Movicon windows, can be '**Docked and kept in foreground view**' as described in the section titled "Docking Project Windows".

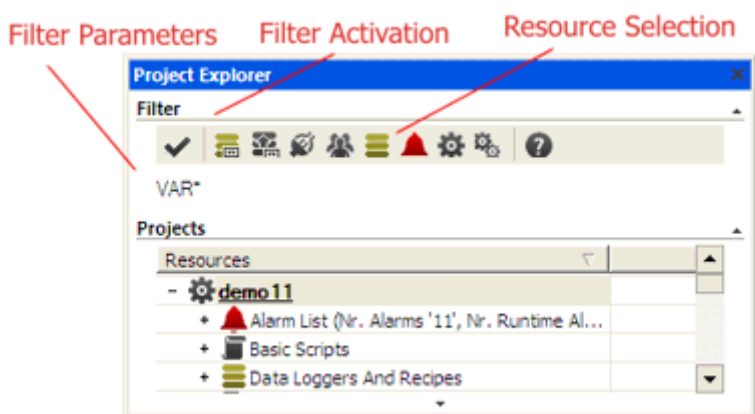


The project structure's items, displayed in the "Project Explorer" window, appear in alphabetical order the first time a resource is explored. when the number of items to be put into alphabetical exceeds the value of 30,000, they will be put into order automatically to save time and bother.

The 'Project Explorer' window divides into three different areas, which can be exploded to display the contents or compressed to hide them. The three areas are "**Filter**", "**Projects**" and "**Commands**".

## Filter

This 'Project Explorer' sub-window is used for creating display filters in elements from various Project Resource groups, displayed in a tree structure in the Project's sub-window as described further ahead:



To execute a filter you must first type the text or the characters to be used as filter in the appropriate box. The "\*" and "?" jolly characters are also allowed. Then you need to select the resource group in which the activate the filter, and this can be done by clicking on the appropriate icon shown in the bar of filters, or by directly selecting the resource group in

question with the mouse in the 'Projects' sub-window. After this you only need to activate the filter with the appropriate button presented in the bar of filters (👉). To delete any active filters just cancel the filter's characters from the box or leave the "\*" jolly character only, then press the filter activation button again.

## Projects

---

This 'Project Explorer' sub-window gathers the project's resources together in groups in a tree structure. Each group is composed of elements which can be added, changed and cancelled by the programmer to get a complete project as required.

## Comands

---

This sub-window in the 'Project Explorer' shows a list of commands which dynamically changed according to the Resource Group selected at that moment:

For example, if you select the "Real Time DB" group the listed commands will be:

- Add a New Variable
- Add a New Comm. Driver
- Add a New Struct Prototype
- Add a System Variable

The same commands can also be accessed by right mouse clicking on the desired element. To check the list of all the available commands see the section titled **"Context Resource Menus"**.

# Properties Window

---

The Movicon **"Properties Window"** is fundamentally important in any project for setting the characteristics of the Movicon resources or components.

Due to the extreme simplicity of the Movicon characteristics the properties window is always used for configuring its resources.

This window is easy to understand and use allowing the characteristics, variables according to the object or resource selected to be set very quickly and easily.

The set properties will be activated immediately after confirmation has been received from the OK icon on the bar at the top of the same window.

The **"Properties Window"** is activated by using the 'Property' Command from the 'View' menu.

Movicon memorizes the status of each property group for each object type. In this way when an object is selected the property groups will be represented with the settings they were given the last time.

## Properties Window Activation

---

There are various ways of displaying the 'Properties Window':

1. Activate the 'Property' command from the Project's Edit menu
2. Right mouse click on the Movicon 'Workspace' and select the 'Property' item
3. Right mouse click on any one of the Movicon Resources or Controls and select the 'Property' item

By using any one of these modes will display the 'Properties Window' relating to the Resource or the Control selected in that moment.

Every time you select a different Resource or Control the 'Properties Window' will automatically refresh its fields different adapting them to the properties of the component being focused on.

The 'Properties Window', as all the Movicon windows, can be **'Docked and kept in foreground view'** as described in the section on ("Docking Project Windows").

## Properties Window Settings

---

The 'Properties Window' of a Resource or Control allow one or more setting modalities according to the type of Resource or Control selected. After having activated the 'Property Window', you can select the Resource's or Control's settings by choosing them from those available in the window.

The main settings concern the Style, Execution or the General settings of the specified Resource or Control. You can also select how to display the list of available properties inside the window as described below:









## Properties Window Displays

---

The display of properties in the 'Properties Window' can be customized by using the command bar at the top of the window:




The icons shown on the command bar mean:

-  Pressing this validation button activates any modifications carried out in the 'Properties Window' in the component selected.
-  Pressing this button deletes any modifications carried out in the 'Properties Window'. In this case only the modifications done after the last validation command execution will be cancelled.
-  Pressing this button displays the properties by group type. For instance the groups may be 'General', 'Style', 'Execution', etc.
-  Pressing this button displays the properties in alphabetical order. In this case subdivisions in group effect will disappear.
-  This button is enabled only when the properties are displayed in groups. In this case by pressing this button you can pass from 'Exploded' to 'Compressed' group displays or vice-versa. This allows you to hide or view all the properties contained in each group.
-  Pressing this button permits you to display or hide the help window which appears at the bottom of the 'Properties Window'. This help window only gives a reduced description string of the property selected. It may be handier to supply the property description immediately and without having to open a supplementary help window.
-  Pressing this button directly opens a Movicon help online file on the top belonging to the selected property description.
-  Pressing this button allows you to pass from the "Advanced" (not easy) to the "Easy" display and viceversa of the properties listed in the window.

## Easy Mode

---

The Properties Window has two display modes. "Easy" and "Normal". The display is changed by using the  bottom from the toolbar in the same window.

In "normal" mode (button released) Movicon displays all the available properties. In "Easy" mode (pressed button) the most advanced and least used properties are hidden from view to simplify user interfacing for the less experienced programmer.

Further more, you can define new property groups and custom colour the property fields as desired. Property window customizing is done using xml files which can be edited as pleased. These files are described below.



The xml files' configuration for customizing the properties window has been purposely designed by Progea based on the most common uses of the product. However, these files can be modified as pleased and are found in the Movicon installation folder. This type of operation is recommended for the expert user only.

#### **"PropVis.xml", "PropVisEasy.xml" and "PropVisEasyPopup.xml.xml" Files**

These three files allow you to decide which properties, or property groups, are to be hidden in the Properties Window during project design mode. All of these files have identical structures. The "property id" attribute is used in the xml files to identify the property through its id in the Movicon resources. To make things easier, this ID also appears in the Help string of each property which appears just by clicking on the interested property so that you can verify it. The "visible" attribute allows you to define whether the property must be visible or not. The property will be visible when set at "1", otherwise it will remain hidden when set at "0".

The settings of the two PropVis.xml e PropVisEasy.xml files are activated with the "Toggle Easy Mode" button. The PropVis.xml file settings will be loaded when the Easy mode is not active, while the PropVisEasy.xml. file settings will be loaded when the Easy mode is active. The PropVisEasyPopup.xml file is used for loading the settings for screen object property Popup windows (click on the top right corner of the object).

#### **".xml" file structure:**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<VisibleState>
<property id="12001" visible="1" description="Crypted Project - Allows you to
save the project file crypted"/>
<property id="12004" visible="1" description="Win2k/WinXP - Enables project
development for Win2K/WinXP"/>
...
...
</VisibleState>
```

The "description" attribute's only purpose is to describe the property being edited, and is not at all taken into consideration by Movicon.

#### **"PropGroup.xml", "PropGroupEasy.xml" e "PropGroupEasyPopup.xml" Files**

These three files allow you to decide which property groups are to be displayed in the Properties Window. By using these files you can define which groups each property is to belong to. Each one of these files has an identical structure. The "property id" attribute is used in the xml files to identify the property through its id in the Movicon resources. To make things easier, this ID also appears in the Help string of each property which appears just by clicking on the interested property so that you can verify it. The two PropGroup.xml and PropGroupEasy.xml file settings are activated with the "Toggle Easy Mode" button. The PropGroup.xml file settings will be loaded when the Easy mode is not active, while the PropGroupEasy.xml. file settings will be loaded when the Easy mode is active. The PropGroupEasyPopup.xml file is used for loading the settings for screen object property Popup windows (click on the top right corner of the object).

#### **".xml" file structure:**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<GroupState>
<property id="12000" groupid="" groupname="" child="0"
description="General"/>
<property id="12001" groupid="" groupname="" child="0" description="Crypted
Project - Allows you to save the project file crypted"/>
<property id="12081" groupid="30011" groupname="Advanced" child="1"
description="Area - Select the variable memory area"/>
<property id="12082" groupid="30011" groupname="Advanced" child="1"
description="Address - Enter the variable address (for bits use the form x.x, for
arrays x (x))"/>
...
...
</GroupState>
```

The "groupid" attribute represents the ID of the group to which the property must belong to. When this attribute is left blank or is the same as its default property's group ID, the property will be associated to its default group and therefore everything will remain the same as before. When the "groupid" attribute is associated with an ID of another group, the property will be moved to the group identified by that ID. When the "groupid" attribute is associated with a new ID not already assigned, a new group will be created in the properties window with the name specified in the "groupname" attribute. This happens only if the "child" attribute is set at "0". However, if the "child" attribute is set at "1", the new group will be created as a subgroup of the property's default group. In this case the subgroup will also have the name defined in the "groupname" attribute. The group IDs created using these files can then be inserted in the PropVis.xml, PropVisEasy.xml and PropVisEasyPopup.xml files for managing their visibilities. The "description" attribute's only purpose is to describe the property being edited, and is not at all taken into consideration by Movcion.

#### **"PropColors.xml", "PropColorsEasy.xml" and "PropColorsEasyPopup.xml" Files**

These three files allow you to pick the back and text colours of the properties displayed in the Properties Window. By using these files you can define each property with a background colour and the colour to be used in the text. Each one of these files has identical structures. The "property id" attribute is used in the xml files to identify the property through its id in the Movicon resources. To make things easier, this ID also appears in the Help string of each property which appears just by clicking on the interested property so that you can verify it. The two PropColors.xml and PropColorsEasy.xml file settings are activated with the "Toggle Easy Mode" button. The PropColors.xml file settings will be loaded when the Easy mode is not active, while the PropColorsEasy.xml file settings will be loaded when the Easy mode is active. The PropColorsEasyPopup.xml file is used for loading the settings for screen object property Popup windows (click on the top right corner of the object).

#### **Struttura dei file ".xml":**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ColorsState>
<property id="12001" back="255" text="16777215" description="Crypted
Project - Allows you to save the project file crypted"/>
<property id="12004" back="255" text="16777215" description="Win2k/WinXP
- Enables project development for Win2K/WinXP"/>
...
...
</ColorsState>
```

The "back" attribute represents the property's backcolor whereas the "text" attribute represents the property's text color. The value is a long number type that identifies the color code.

The "description" attribute's only purpose is to describe the property being edited, and is not at all taken into consideration by Movcion.

## **Logic Explorer Window**

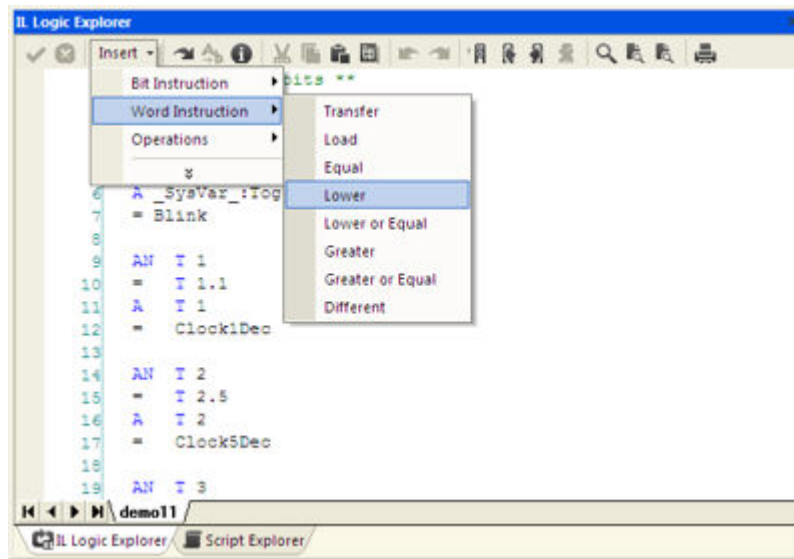
---

The '**Logic Explorer**' Window is of fundamental importance for editing the project's '**IL Logic**'. This window, acting as a text editor, allows you to insert IL logic codes. The window's contents change dynamically according to the project's object or resource currently being selected. Codes can be associated in IL Logic format either to the project (General Logic) or to each single Movicon symbol or drawing (Local Logic). Therefore by selecting a symbol, any inserted code relating to that symbol will appear in the 'Logic Explorer' window.

Typing in code has been made easy due to the fact that the instruction syntaxes are automatically acknowledged and represented with different colours. For instance the instructions are represented in blue while the comments are displayed in green. The rest remain in black.



The insertion of the instructions can also be executed by using the appropriate 'Insert' menu from the 'Logic Explorer' window. This is very handy especially when the exact syntax of the instructions to be used is not known.























In addition to this, the F8 key has also been provided for calling the "Tag Browser" window, which is very handy to have when inserting variables.

You can execute the editing commands and run checks on the logic by using the command bar positioned at the top of the 'Logic Explorer' window:



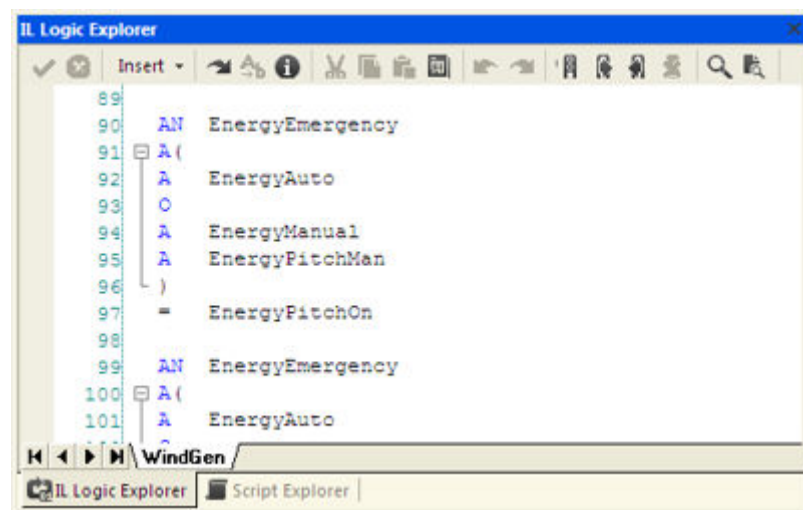
The icons presented in the command bar are:

-  Pressing this OK button will activate any changes made in the 'Logic Explorer' window.
-  Pressing this button will delete any changes carried out in the 'Logic Explorer' window. In this case only the changes done after the last OK button command execution are deleted.
-  Pressing this button will execute a control of any syntax or instruction errors inserted in the code. Any errors found will be described in the Movicon status bar.
-  Pressing this button will consent to the automatic completion of the instructions being inserted.
-  
-  Cuts the selected text. In this case the text is cancelled but kept in memory on the Windows Clipboard.
-  Copies the text selected to the Windows Clipboard.
-  Pastes the Windows Clipboard's contents on the pointed indicated by the cursor.
-  Selects all the texts contained in the 'Logic Explorer' window.

-  reverses the last operation executed (ie. cancelling of a text, etc.).
-  restores the last operation Cancelled (ie. cancelling of a text, etc).
-  bookmark
-  toggle bookmark
-  next bookmark
-  cancel bookmark
-  Find
-  Replace
-  Find Next
-  This button is used for getting a printout of the Logic text. The print selection settings window will appear.

## Objects being Edited

When a component is selected, its name is shown in the bar at the bottom of the window. This helps you understand which component is being referred to by the logic displayed in the window:



# ToolBox Window

---

In the Movicon **'ToolBox'** window you will find many types of drawings and controls to pick and insert on screen. These objects have been grouped into five different categories according to how they work.

The **'ToolBox'** can be opened using the 'Toolbar' command from the 'View' menu.

All the Drawings and Controls are vectorial type objects and have some properties in common such as the Animation properties.



The Movicon Drawings and Controls can be grouped together to form composed symbols.

## Basic Shapes

In this category you will find a series of drawings, or simple geometric shapes, which are mainly dedicated to create the screen's graphical design. These components do not have properties specified for Command executions but they support all the animation properties common to all the other Movicon vectorial components.



Even though they do not have specific properties for executing command, they can, however, execute Commands by being edited with Script codes internally.

## Buttons-Lights-Switches

In this category you will find a series of **"Buttons"** type controls which are dedicated to executing the Commands defined in the **"Command List"** or simple variable settings. This category includes list of various types of objects such as Coloured Lights or Coloured Buttons which all derived from the same Button component modified in the Style Properties.

## Sliders-Gauges-Meters-Displays

In this category you will find a series of **"Gauge"** type controls in all sorts of graphical shapes. There are a list of different objects which all derive from the same Gauge component modified in the Style properties. In addition to this you will also find an "Editbox - Display" object.

## Trend-Charts-Data Analysis

In this category You will find a set of **"Trend"** controls and the Chart object. The purpose of these controls is to display variable behaviour graphically.

## Advanced Shapes

In this category you will find a series of controls of diverse types and functionalities. These include types such as Displays, Alarm Viewers, Historical Log and Data Logger Viewers, Charts, etc.

## Custom ToolBox Categories

---

New Symbols can be created added to the Toolbox as new categories. These symbols can also be inserted into the Movicon "Symbols Library" to create new library categories. In order to do this, you first need to create, if not already done so, a Movicon Installation subfolder in the name of "ToolBox". Then create a category in the "Symbols Libraries", containing the symbols of interest and copy the created symbol libraries ".msxz" file inside the "ToolBox" folder. The new symbols will then be displayed within the ToolBox as a new category with the same name of the ".msxz" file and containing the symbols inserted in the ".msxz" file.

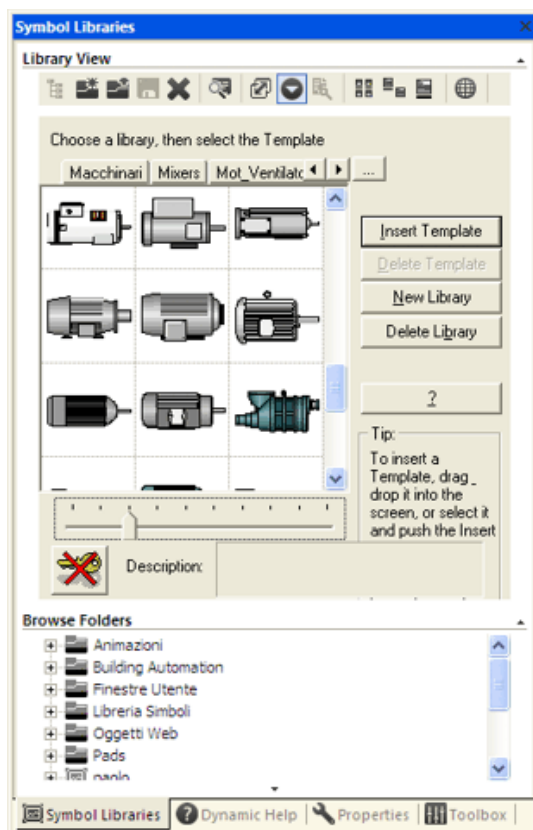
This will allow you to extend the "ToolBox" contents with customized symbols rendering them available for immediate use without having to get them from the "Symbol Libraries".

# Symbols Library Window

The graphics from the library can be customized and used in Movicon vectorial drawings. The system provides you with a pre-built library of standard graphic symbols which can be accessed through the '**Symbols Library**', and used as vectorial drawings by the programmer in editing screens.

The '**Symbols Library**' can be modified by the same programmer who can create his/her own symbols and insert them into the library which can be customized and made bigger.

The '**Symbols Library**' can be activated with the 'Symbols Library' command from the 'View' menu.



The Symbol library can be displayed by category or in a tree structure.



This current list of symbols and categories may be subject to changes without prior warning but can be customized by the user to create new symbols or to change existing ones.

The symbols are subdivided into categories in the library. This window is divided into two panes, the top pane shows the objects within the selected category, while the bottom pane shows existing categories in a tree structure. When clicking on one of the categories in the tree structure, the top pane will automatically update and show the objects of the category selected. These categories can be browsed through by clicking on the relative tabs or using the scroll buttons located on the top right of the top window pane.

You can create new categories or delete them by using the appropriate buttons placed on the window's right border.

Buttons	Description
<b>Insert Template</b>	This command executes the insertion of the symbol selected in the library in the current screen.
<b>Delete Template</b>	This command deletes the symbol selected in the library.
<b>New Category</b>	This command allows a new category to be added to the library. Movicon will request the name of the new library.
<b>Delete Category</b>	This command deletes the Category selected in the library.
<b>?</b>	This command calls up the selected symbol's or category's help.

A category's symbols can be inspected by using the side scroll bar. In addition to this you can also use the cursor placed on the bottom preview window to zoom in and out of the symbol. You can customize the window display by using the command bar positioned at the top of the 'Symbols Library':



The icons presented in the command bar are described below:



Command not available.



Pressing this button adds a new category to the library. Movicon will request you to insert the name of the new category.



Pressing this button allows a category to be opened by selecting it directly through the Windows file browser window.



Pressing this button executes a library save. This button is only active after the library has been modified.



Pressing this button deletes the category selected in the library.



This button executes a search for the specified category.



This button is used for displaying or hiding the button on the right hand side of the window. By doing this only the symbols will be seen in the window.



This button is used for activating or deactivating the symbol's animation preview.



Pressing this button opens the **"Dynamic Property Inspector"** window relating to the selected symbol.



By using these three buttons you can select three different symbol display sizes. You can also use the bar at the bottom of the window to get a more precise symbol size.



Pressing this button executes a search on Progea's website for any symbol library updating. This can only be done when the PC is connected to Internet.



Pressing this button calls up the selected symbol's or category's help.

## Symbol Library Protection

---

The 'Symbol Library' can be given password protection to safeguard any customization carried out. To activate the symbol's password protection you need to press the button on the bottom left border representing a key. The system will request you to type in the protection password, after which the library will be protected. To deactivate this protection repeat the same procedure. The protection button icon will declare whether the protection is active or not:



Locked Symbol Category.

When the symbols window shows this icon, it means that the symbols of the selected category have been locked and cannot be ungrouped or exported. To unlock the category click on the icon and re-enter the protection password.



Unlocked Symbol Category.

When the symbols window shows this icon, it means that the symbols in the selected category have been unlocked and freed. To protect the category, click on the icon and enter the protection password.

## Favourite Categories

---

The 'Symbols Library' contains a large series of categories, listed in alphabetical order. The scroll arrows and the [...] search button are used for a much easier search of the symbol categories desired.

You may find it handier to use the 'Favourite Categories' which allows you to show the most frequently used categories on the left hand side of the library to point them out quicker.

To label a symbol category as 'Favourite', right mouse click on the category of interest, then execute the 'Add to Favourites' command that displays ?.

The Tab relating to the selected category will appear on the left hand side to evidence it from the rest.

To delete one of your Favourite categories, carry out the same procedure by selecting the **'Remove from Favourites'** command instead.

# Dynamic Property Inspector Window

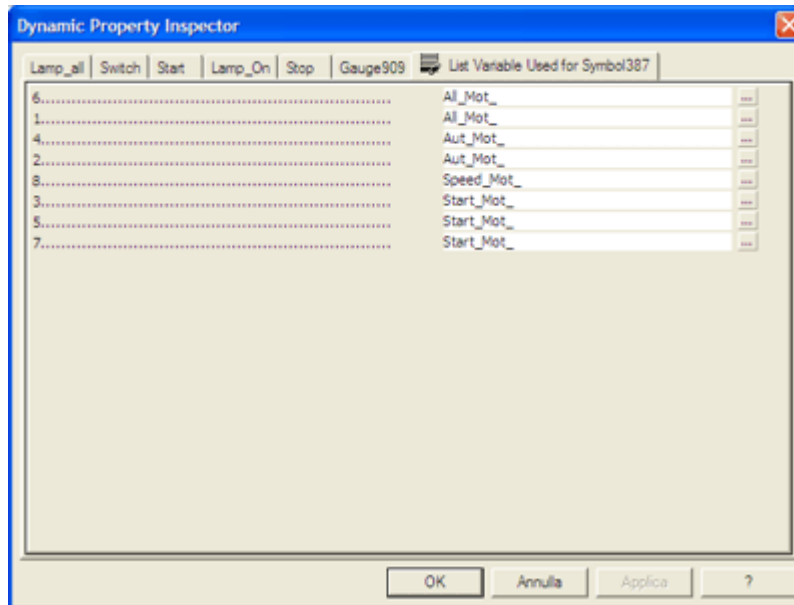
The '**Dynamic Property Inspector Window**' is a very handy tool for the programmer to have when editing symbols.

This tool consists of a inspection window through which you can examine the whole structure of the variable associated to the dynamic property and any basic script codes existing in the symbol.

The '**Dynamic Property Inspector Window**' is activated by selecting the symbol of interest with the appropriate command which appears when clicking the right mouse key.



This function is not available when the symbol has been withdrawn from a symbol category protected with a password in the library.



By using the symbol's 'Dynamic Property Inspector' window you can view the symbol's structure and directly access all the variables used and the basic script codes contained in the symbol's elements.

Each one of the Tabs, at the top to the window, represents each single component of the symbol. When clicking on the desired symbol element the dynamic property inspector will display the variables associated to the symbol's execution dynamic properties and any script codes.



The '**Dynamic Property Inspector**' window normally is presented every time a symbol from the "Symbols Library" is inserted containing animation variables or basic codes. In this case the window will open on the "List Variables Used for..." tab where all the variables used within the symbol are be listed.

## Replacing Variables

The inspector window indicates the variables which the selected drawing element has been associated with by specifying the interested dynamic property. Each one of the variables can be replaced with a different one, or the Variables DB can be accessed by clicking on the [...] setting button on the right of the Variable's box. By doing this you can directly assign a replacement variable existing in the DB or create a new one, which will be introduced in the Variables DB and assigned to the selected element to replace the one previously associated to the symbol.

## Editing Basic Scripts

The inspector window will show any base codes associated to the selected drawing element.

The basic code can be displayed and changed directly from the inspector window. The changes made will be assigned to the drawing object and the new code will replace the old one.



You may also find it very handy to use the F8 key to call the "Tag Browser" window when inserting variables.

By using the same window you can select the events which each symbol element is to dispose in function with its characteristics.

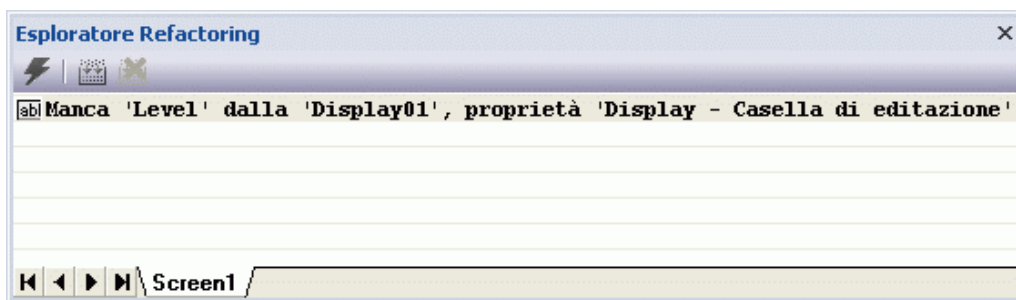


# Refactoring Explorer Window

The "Refactoring Explorer" window, once displayed, allows you to view any errors that may have arisen for the resource you selected. The errors refer to variables inserted in the properties of the resources and objects which have not been defined in the Real Time DB, or can be due to objects placement out of the screen area (see "Outside Screen Objects"). This control is active for all the project's resources which can be associated variables. By selecting a resource, i.e. a Screen, the Refactoring Explorer window will be populated with the errors found for that screen or for the objects it contains. Error messages already give explicative details on the problem verified. By double-clicking on the error line in cased concerning objects in screen errors, the screen will open with focus on the object in question, along with the "Dynamic Property Inspector" window.



**The variables used in the basic script code are taken from the Refactoring Explorer window. Only the "Status Variables" associated in the basic script execution properties can be taken from the Refactoring Explorer window.**



By using the command bar place at the top of the "Refactoring Explorer" window, you can execute rebuild commands for the resource you selected:



The function of the icons shown in this command bar are:



**Open:** pressing this button will open the screen where the error was found. When the selected resource is not a screen, this command will have no effect.



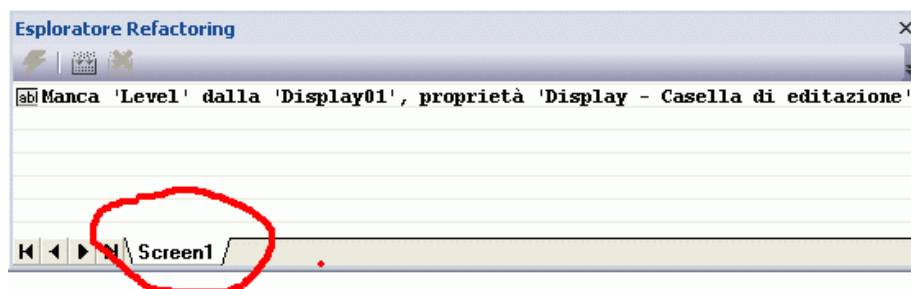
**Rebuild:** pressing this button will rebuild the list of errors for the selected resource.



**Stop build:** pressing this button will stop the rebuilding of the error list which was requested with the "Rebuild" button.

## Editing Objects

When selecting an item, its name will appear in the window's bottom bar to let you know which component type is being referred to in the error list displayed in the window:

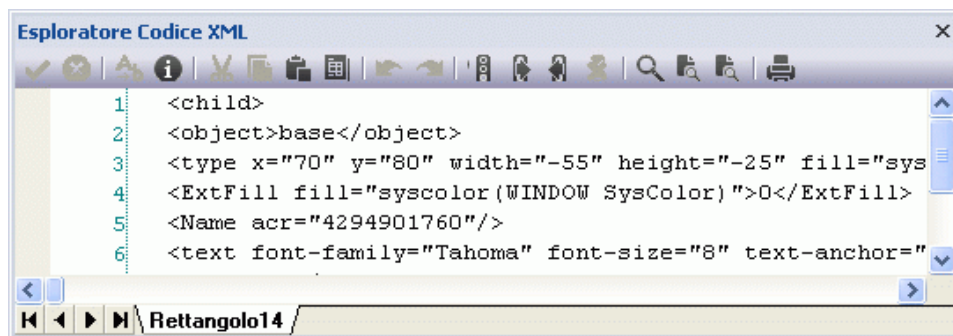


## XML Code Explorer Window

The "XML Code Explorer", once made visible, consents you to view the xml codes relating to the resource or object selected. Apart from viewing object xml codes, this window can also be used to modify them. These modifications will then be applied graphically to the object once confirmed while the object's properties will only be updated the next time it is selected.



**The selected object's XML code can be modified directly through this window. However, you will need to know the right syntax to use and what the XML tags mean.**



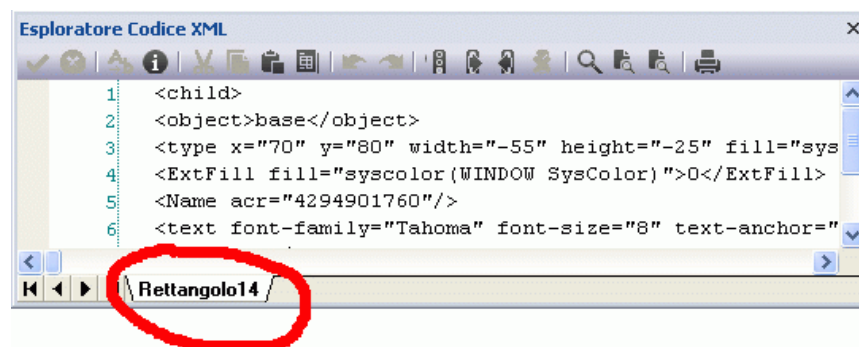
You can execute XML code editing commands by using the command bar place at the top of the "XML Code Explorer":



The commands provided are for editing text, copy&paste, find, etc.

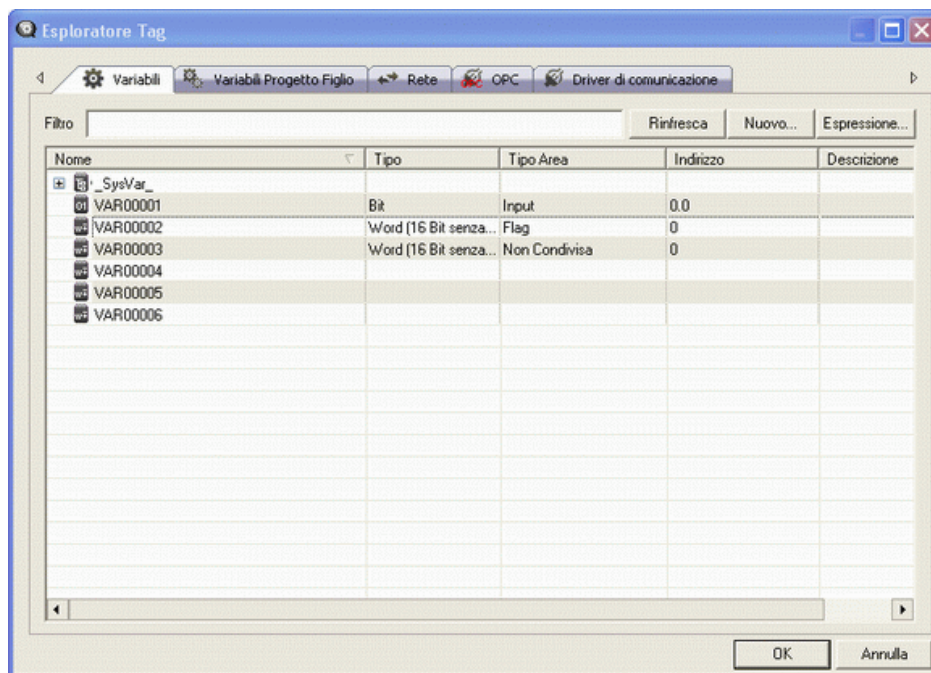
### Editing Objects

When selecting an item, its name will be displayed in the window's bottom bar. This lets you know which item the XML code, displayed in the window, refers to:



# Tag Browser Window

The "Tag Browser" window can be opened with the "..." button in all those object or resource properties where variables or dynamic links can be inserted. This window provides the list of all the project's variables and dynamic links (OPC, Networking and Communication Drivers) which can be selected and then inserted in the object and resource properties.



The "Tag Explorer" window can also be opened with the F8 key when in the following windows for editing script codes or IL logic:

- Script Explorer Window
- Logic Explorer Window
- Dynamic Property Inspector Window

The "Tag Explorer" window opens upon pressing the F8 key and after having selected the variable, focus will return to the point in which the variable was inserted in the code. Using the F8 cuts down on the number clicks/operations usually needed for adding variables to code.



The "Tag Explorer" window cannot be opened with the F8 key from a Basic Script Edit window where it functions in a different way within this context.

The types of selections available on the Tag Browser's Tabs are:

## Variables

This window shows the list of the variables available in the Real Time DB for the project currently opened. The command buttons presented are:

**Refresh:** refreshes the list taking into account any filter conditions inserted in the "Filter" field

**New...:** inserts a new variable in the Real Time DB by opening a window where you can also edit the new variable's properties

**Expression...:** opens a VBA Expression editor to allow you to insert a "Basic Script Expressions" (this will replace the single variable)

Variable properties can be edited by using a window which opens with a right mouse click on the variable you wish to edit.

#### Child Project Variables

This window shows the list of variables available in the Real Time DB of eventual child projects. The 'Refresh' and 'New' command buttons are provided and have the same functions as the buttons provided in the "Variables" group but refer to the child project's Real Time DB.

#### Network

This window consents you to select a project locally or from a PC in network and displays a list of the variables which the project has been provided with for networking.

#### OPC

This window consents you to select a local OPC Server or one from PC in network and displays the list of available tags for the Client.

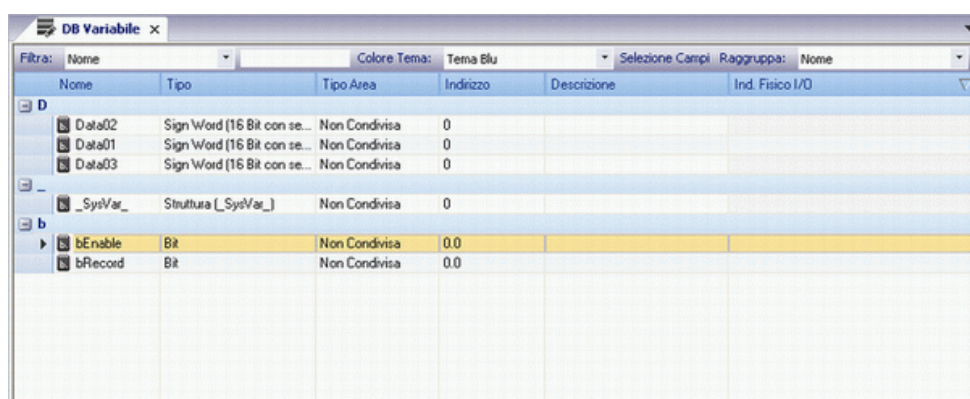
#### Comm. I/O Drivers

This window consents you to add dynamic links to the Communication Driver. The Drivers inserted in the project will be displayed on the list and Tasks can be inserted by double-clicking on the driver of interest.

## List Variables Window

Movicon has a variable editor in table form, which can be opened from the Project Explorer window and consents you to set and search for variables much faster. This table can be presented in two versions:

- A very fast version, displayed purely in table form like the one used for editing project strings. This version allows you to open a database composed of many variables (even more than 50,000) in less than one second. Filtering or grouping cannot be performed on this table.
- Not so fast version but consents variable filtering, ordering and grouping.



The screenshot shows a window titled 'DB Variabile' with a table of variables. The table has columns: Nome, Tipo, Tipo Area, Indirizzo, Descrizione, and Ind. Fisico I/O. The table is filtered to show variables starting with 'D' and 'b'. The variables listed are Data02, Data01, Data03, \_SysVar\_, bEnable, and bRecord.

Nome	Tipo	Tipo Area	Indirizzo	Descrizione	Ind. Fisico I/O
Data02	Sign Word (16 Bit con se...	Non Condivisa	0		
Data01	Sign Word (16 Bit con se...	Non Condivisa	0		
Data03	Sign Word (16 Bit con se...	Non Condivisa	0		
_SysVar_	Struttura (_SysVar_)	Non Condivisa	0		
bEnable	Bit	Non Condivisa	0.0		
bRecord	Bit	Non Condivisa	0.0		

This variable can be opened with a double-click on the "List Variables (Tags)" group or on the name of a variable group in the Project Explorer window. If the project contains more than 3,000 variables, Movicon will ask you if you wish to use the less powerful but extremely faster editor instead.

In any case it is always possible to force the faster editor open by pressing the keyboard "CTRL" key while double-clicking with the mouse.

By using the new editor you will find it much faster to put into order and filter the variables compared to just using the project Explorer window.

This RealTimeDB Variables table's only purpose is to very clearly and quickly display variable lists, but it does not allow you to carry out any modifications directly in it. The properties of one or more selected variables can be changed using the Properties Window.

The list can be put into order by its item types by simply clicking the column's title. The commands for applying filters or grouping variables are at the top of the window and can be:

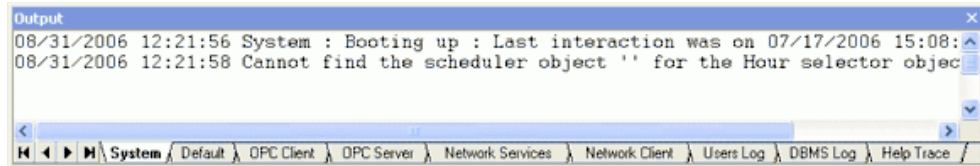
- **Filter:** this list-box allows you to select whether to filter by "Name", "Description" or "Dynamic". In the entry field at the side of the list you can in fact enter the filter type to be applied
- **Color Theme:** this list-box allows you to select the color to apply to the table. The choice of colors are light blue, silver and Green
- **Field Chooser:** not available for the time being
- **Group By:** this list-box allows you to select how to group the variables. The current choices by "Type", "Area", "Address" and "Name"



When changing the properties of variables, they may appear in the wrong groups until regrouped again. This functionality has been purposely designed to give priority to quicker editing when dealing with large amounts of variables.

# Output Window

The Movicon **"Output"** window is used for displaying system and error messages which are generated by Movicon during the development or Runtime phase. These messages are also saved on log files during the Runtime phase as usual, but by using this window you can get a more immediate view of the situation which is especially handy in the project debug phase.



You can select different Tabs from the bottom of the window for viewing messages according to the different groupings as described below.



When right clicking on the area of the window a menu will appear with the **"Clear all messages"** command. When this command is confirmed all the active TAB messages will be cancelled, but only those from the Output window and the Historical Log window.

## System

Lists the system messages, such as project run, driver communication status, etc.

## Default

Lists the Project's default messages.

## OPC Server

Lists the messages inherent in the Movicon OPC Server.

## OPC Client

Lists the messages inherent in the Movicon OPC Client.

## Networking Services

Lists the messages inherent in the Movicon Networking Server messages.

## Networking Client

Lists the messages inherent in the Movicon Networking Client messages.

## Users Log

Lists the messages inherent in the Project's Users Log.

## DBMS Log

Messages, inherent to log recordings and executed through the ODBC or IMDB, are listed.

## Trace Help

Returns the index number of the Movicon dynamic help required.

## Basic Script Messages

When the Basic Script's "Create its Tab Trace" property is enabled, a new TAB will be created with the same name of the Basic Script in runtime where the Debug.Print messages, in the Basic code, will be printed.

# Dynamic Help Window

---

The Movicon '**Dynamic Help**' window makes it easier to search the guide for the argument relating to the selected resource, component or symbol. The difference between this and the usual Help, which opens the specific topic on the selected argument only, is that this window shows different links to arguments linked to the main topic in order to give the programmer a wider panorama of information.

Topics are shown in groups in the Dynamic Help window some of which change every time according to the object selected, while others are linked to general or frequently consulted arguments:

- **Linked Topics:** are a series of links which change dynamically according to the Topic which has been selected.
- **General Information:** is a series of links concerning the general or frequently consulted information/asked questions.
- **Technical Support:** is a series of links regarding the Movicon technical support.

# Watch Window

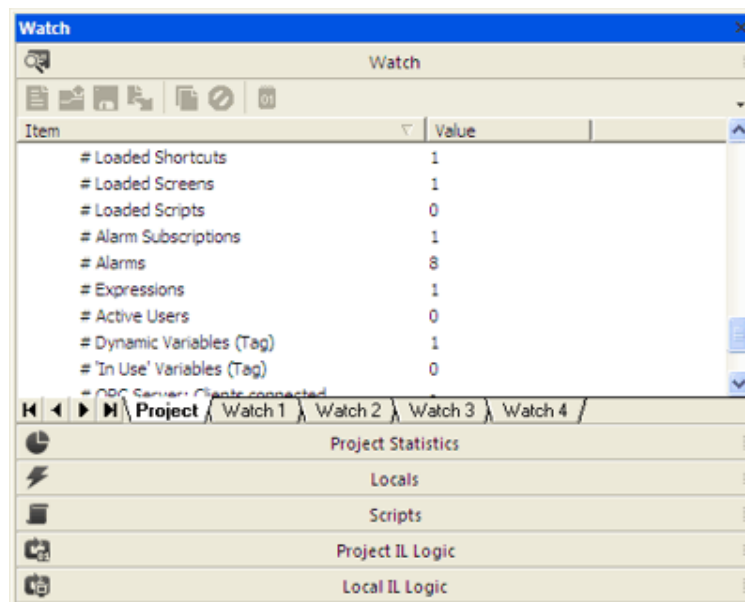
The Movicon **"Watch"** window is used for displaying the status of the project, logic debugging and forcing variables during the project Runtime phase. This window is only available when the project's Runtime is started up from the development session. The **"Watch"** window is composed of groups, each one displays certain information:

## Watch

The Watch group is composed of different TABs. The 'Project' TAB contains information on the project's general characteristics, such as the number of variables in use, the number of active screens, the number of connected Clients, etc. The other TABs, called 'Watch 1', 'Watch 2', etc., allow one or more variables to be selected for monitoring or changing.



**In order to carry out any changes to a variable, simply click the right side of the displayed value and then enter a new one. As an alternative, after having selected the line with the variable, press down the F3 key to enable value editing.**



When opening the Tag list for inserting a new variable in the Watch window, Movicon will open a quick variable editor (List Variables Window) within a dialog window if there are more than 500 variables existing in the project. This dialog window has been purposely implemented to save the user time which otherwise would take longer when there are a lot of variables in the Real Time DB. The only thing you cannot do when using this edit dialog window is to select single members from a structure variable therefore the whole structure variable must be selected.

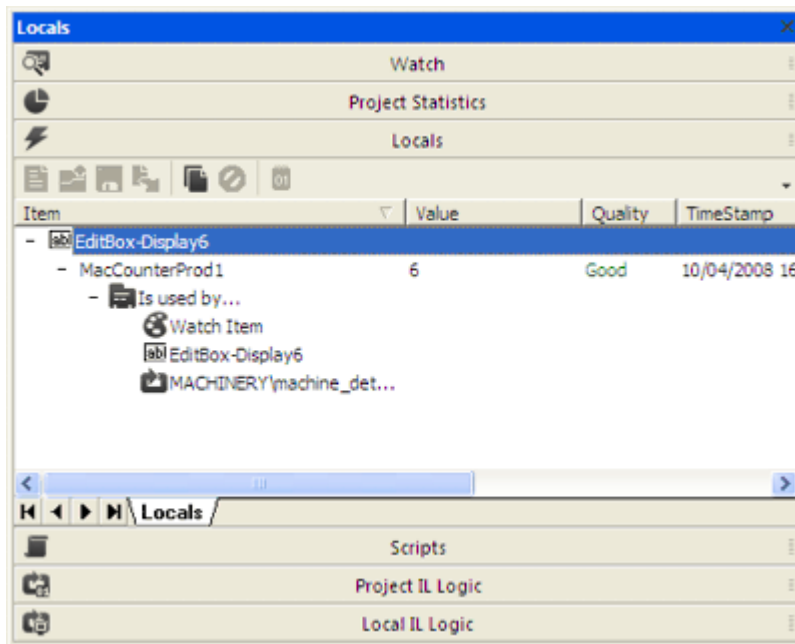
## Project Statistics

The Project Statistics group documents the use of the resources retrieved by the system by representing them on a table correlated with pie charts.

## Locals

This Locals group allows information on the on screen components to be displayed. When clicking on an on screen object, the window will change its contents by displaying information based on that component.

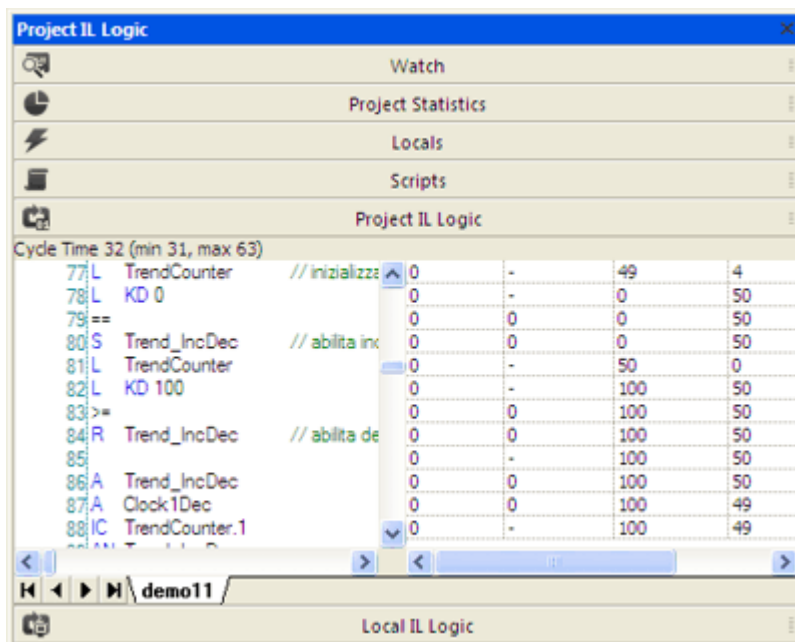




When double-clicking on an object in this window containing Basic Script code, its Basic Script code debug window will open. When the object does not contain code, a beep will sound and no window will open.

## IL General Logic

The IL General Logic group allows the Debug of the IL Logic resource associated to the project to be run.



## IL Local Logic

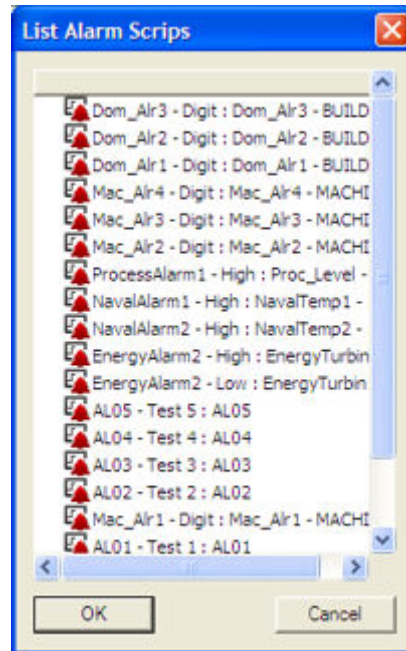
The IL Local Logic group allows the Debug of the IL Log associated to the component or screen selected with the mouse to be run.

## Script

the Script group lists the Basic Script resources loaded in memory with some useful information such as status (running or stop) and total execution time of last run or the run taking place. When double-clicking on the name of a listed Basic Script, the debug window of that basic script resource will open.

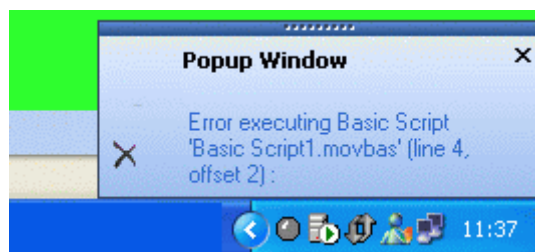
## Alarm Script Debugger...

This window is opened by using the "Alarm Script Debugger..." command from the "RunTime Bar". This window lists all the alarms managed in the project. When selecting one of these alarms, the debug window of the Basic Script code inserted in the alarm's threshold will open.



## Popup Window

The Popup Window appears near the system icon when Movicon opens a non modal error window during project Runtime. This Popup Window is used to allow the error window to open when it has been put in background in respect to the Movicon window.



# The Resources

---

The **Resources** form the foundations of the Movicon applied project. Each project gathers all the Resources by organizing them into **Groups**, based on function types, which are listed in the "**Project Explorer**" window. This window, in tree structure, is the container of all the project's resources, and therefore the project itself. Resources can be accessed from the 'Project Explorer' window for configuring properties, or inserting new resources or deleting them from the project.

## Inserting a new Resource

---

To add new Resources to the project, position the mouse on the Group where you wish to add an item, then right mouse click and select one of the items available from the menu which opens. The same list of options can also be accessed from the "**Commands**" window of the 'Project Explorer's' window.



The list of commands displayed in the menu opened with the right mouse key or in the 'Commands' window changes dynamically according to the Group or Resource being selected.

## Editing Resources

---

To edit a Resource you only need to select it and open the "**Properties Window**". By using the "**Properties Window**" you can change the settings in the properties of each Resource. The "Screens" and "Basic Script" resources are the only ones that need opening with a double-click.

## Deleting Resources

---

Before deleting a resource you need to select it first. The resource is deleted by using the CANC key or the "Cut" or "Delete" commands from the "Edit" menu. A command executed in error can be cancelled by using the "Undo" command from the "Edit" menu or the CTRL+Z keys. You can delete the Resources inserted within groups but you cannot delete the "Resource Groups" listed in the "Project Explorer" window created for default by Movicon.

## Cut, Copy and Paste Resources

---

Each resource can be cut, copied and pasted within the same project or in different projects. Before carrying out any one of these commands you need to select the resource or resources first.

The Cut, Copy and Paste command can be accessed from the "Edit" menu by using the right mouse key or from the keyboard using the CTRL+X, CTRL+C and CTRL+V keys respectively. These commands can also be carried out with multiple resource selections. To select more than one resource keep the CTRL key pressed down and click on the resources desired or keep the SHIFT key pressed down and select the resources with the UP/DOWN arrow keys (also see the "Cut, Copy, Paste" paragraph in the "Standard Editing Techniques" section).

A command executed by mistake can be cancelled with the "Undo" command from the "Edit" menu or with the CTRL+Z keys.



The Resources can be pasted inside the same Group it belongs to only. For instance, a variable can be pasted inside the "Real Time DB - Variable List (Tag)" Group only and not inside the "Alarm List" or "DataLogger & Recipe" Groups etc..

## The Multiple Project Resources

---

The project's Multiple Resources are those which are saved in a XML file for each inserted resource. These xml files are found in the project's "RESOURCES" folder. There is a group in the "**Project Explorer**" window which identifies each of these resource categories. The resources and consequent relating groups are:

**Screens**  
**Menu**  
**Accelerators**  
**Basic Script**  
**Parameter Files**

Within each of these five groups of resources you can create other folders to simply organize your project better. It may be very handy to group these resources in different Folders in a logical order by keeping different resources which have the same requisites in common (such as belong to same area) grouped together. For instance, supposing you need to create a project for monitoring a plant divided into different zones, you could create one folder for each plant zone and then group in each folder the Screen, menu sources etc, concerning that zone. To get the same folder available in all five groups of the multiple resources you will need to create the folder five times over, meaning that you have to create a folder for each one of the five groups. The folder in reality results as one only. In the project structure displayable through the Windows Resource Explorer window where all the files of the five resource groups, inserted in this folder, will be saved.

This folder must be empty before it can be cancelled and therefore you must check that it is also empty for all the five multiple resource categories, listed above, before doing so.

During the development phase, only those folders containing just one type of multiple resource category can be renamed, otherwise an error will show: "This folder is shared across multiple resource types and cannot be renamed".



A folder used in a resource group can not be renamed using a folder name already in use in another resources group or in the resource group itself.



The only purpose of creating tree structures within the multiple resource groups in the created custom folders, is to organize the project better and do not influence how the project works in any way.

## Common General Resource Properties

---

By using the General properties you can set the name of the resource selected in the "Project Explorer" window. In order to do this just select the Resource desired and then change its settings through the Movicon **"Properties Window"**.

### Name

This edit box lets you enter the name of the selected resource. You can also change the text through the "Project Explorer" window by clicking the resource and pressing the F2 key.

## Resource Folder General Properties

---

The name of the folder inserted in the 'Project Explorer' window's Multiple Resource groups can be defined using its General properties. This is done by selecting the Folder desired and then modifying its settings through the Movicon **"Properties Window"**.

### Folder Name

The name of the folder is entered in this edit box. The text can also be changed through the 'Project Explorer' window by clicking the folder and pressing the F2 key.

### Folder Path

This text box is in read only and shows the folder's complete path.

### Filter

This edit box is used for inserting a string for executing filters on files contained in the folder. The "\*" (star) and "?" (question mark) wild card characters can also be used.

# Project Files

---

Movicon executes the saving of project files in one unique folder, commonly called **'Project Folder'**. In addition to project files, sub-folders containing data relating to the project are also created so that the project's architecture is structured clearer and readable.



All the Movicon project files are saved in XML format, and therefore can be read and edited with any text editor or XML editor. This can only be done if the project files have not been crypted by the programmer.

The data and information of the Movicon projects are contained in files with '.movprj' extensions. Once the project is saved, Movicon will always carry out a backup of the project file before saving it again. The back-up copy will have the same name as the project's but with the 'movbak' extension.

If for any reason the project file is lost, just simply open the backup file (renaming the extension from '.movbak' to '.movprj').

Movicon also associates other files to the project, one for each Resource type. The files will be created in the same project folder.

The following table describes the types of files identified by their extensions:

File Extension	Description
.movprj	Project file
.movbak	Back-up copy of project file
.movalr	File containing the data and the configurations inherent to the 'Alarms List' Resource
.movdlrec	File containing the data and the configurations inherent to the 'Data Loggers & Recipe' Resource
.movevt	File containing the data and the configurations inherent to the 'Event Objects List' Resource
.movils	File containing the "IL General Logic" code of the project
.movnetclt	File containing the data and configurations inherent to the Client 'Network Services' Resource
.movnetsvr	File containing the data and configurations inherent to the Server 'Network Services' Resource
.movopcclient	File containing the data and configurations inherent to the OPC Client
.movrealtimedb	File containing the data and configurations inherent to the 'Real Time DB' Resource
.movsch	File containing the data and the configurations inherent to the 'Scheduler Object List' Resource
.movscl	File containing the data and configurations inherent to the 'Scaling Object List' Resource

string.language	File containing the Movicon text strings in the language specified in the file extension. If, for instance, the Italian and English language have been inserted into the Movicon String Table defining two columns as 'Italian' and 'English', this will create two files as follows:  ProjectNamestring.Italian ProjectNamestring.English
string.language.bak	Backup file of files containing Movicon text strings
.actstr	File containing the actual language column name.
_c	Some files have the above described description plus the "_c". For example:  ProjectName.movalr_c  These are compiles files which are used by Movicon in Runtime mode for enhancing high speed performances of the project being run. These are the same XML files compile in binary format. These compiled files are automatically generated by Movicon at runtime and are re-created each time the related project file is modified.  <b>Warning:</b> directly changing an "_c" compiled file can cause unpredictable behaviour in Movicon, according to the resource type the file refers to. Changing the compiled files for the Data Logger/Recipes, Event objects and Alarms (as an example with an external tool like Notepad), these files will not be re-created at the following start of the project and the project will be executed with the changes made in the compiled files. In case of other resource types, like the screens, if a compiled file differs from the project xml file, the "_c" file will be re-created anyway.
.uxp	This file is created when a user password change is executed. Typically when the property is enabled to force password change at first login or when passwords expire.
.zip	Project Backup File.

As already mentioned above, in addition to the project files, Movicon also creates some folders which will contain information relating, above all, to the saving and recording of data during the runtime phase:

Folder Name	Description
ALARM	Files with the '.alr' extension which contain information on alarms in runtime are inserted into this folder. A file will be created for each project alarm with the following syntax:  <i>ProjectName_ThresholdName_VariableName.alr</i>
DATA	In this folder you can find the following file types:  <b>".var"</b> Contains information relating to the variables declared as

	<p>retentive are inserted into this folder. A file will created For each retentive variable in the project with the following syntax:</p> <p><i>ProjecName_t_VariableName.var</i></p> <p><b>".watch"</b> Contains the configuration information of the Watch window in the project's debug Runtime phase. These files will be named:</p> <p><i>Watch 1.watch</i> <i>Watch 2.watch</i> <i>Watch 3.watch</i> <i>Watch 4.watch</i></p> <p><b>".mdb"</b> The Database file can be found for the <b>"Real Time DBMS Settings"</b> management. The file will be created in this folder for default with the name:</p> <p><i>ProjectName_RealTimeDB.mdb</i></p> <p>You can also find the Database file for the managing of <b>"Variable Trace"</b>. this will be created only when data recording is activated through the ODBC and not the IMDB. The file will be created for default in this file with the name:</p> <p><i>ProjectName_TraceDB.mdb</i></p> <p><b>".dat"</b> You can find this file type for managing the <b>"Variable Trace"</b>. This will only be created when data recording is activated through the IMDB and not the ODBC. The file is in text format but can also be crypted when desired. A file will be created for each variable being traced with the name:</p> <p><i>VariableName.dat</i></p> <p><b>".xml"</b> You can find this type of file for managing the <b>"Variable Trace"</b>. This will be created only when the data recording is activated in the IMDB and not the ODBC. the file is in xml text format. A file will be created for each variable being traced with the name:</p> <p><i>VariableName.xml</i></p>
DLOGGERS	<p>In this folder you can find the following file types:</p> <p><b>".mdb"</b> Here you will find the database file containing the project Data Loggers and Recipe tables. This file will only exist if the selected format is MS Access, as regards to the SQL Server, the Database will be created in the MSDE folder. Furthermore, this will only be created when the data recording is activated in the ODBC and not the IMDB. The file will be named:</p> <p><i>ProjectName_DLR.mdb</i></p> <p><b>".dat"</b> Here you will find the file for managing the project's Data Loggers and Recipes. This file will only be created when the data recording is activated with the IMDB and not</p>

	<p>with ODBC. The file will be in text format but can also be crypted if desired. A file will be created for each Datalogger and Recipe with the name:</p> <p><i>DataLoggerName.dat</i> <i>RecipeName.dat</i></p> <p><b>".xml"</b> You can find this file type for managing the project's Data Loggers and Recipes. This file will only be created when the data recording is activated with the IMDB and not ODBC. This file will be in xml text format. A file will be created for each Datalogger and Recipe with the name:</p> <p><i>DataLoggerName.xml</i> <i>RecipeName.xml</i></p>
LOGS	<p>A few log files, in ASCII format are presented in this folder which report information relating to the different project components, such as system messages, OPC messages, user activation messages, etc. All these files have the '.log' extension and their names indicate their contents type.</p> <p>This file, in Database MS Access format, may be found in this folder, containing data relating to the project's Historical Log. This will only be created when the data recording is activated through the IMDB and not the ODBC. The file will have the default name of:</p> <p><i>ProjectName_HisLog.mdb</i></p> <p>If however, the data recording is set through the IMDB and not the ODBC there will be files with the ".dat" extension and ".xml" containing data relating to the Historical Log. The ".dat" files are in text format but can be crypted.</p>
NETLOG	<p>Some log files, in ASCII format, are in this folder reporting information relating to the Networking communications. The information relating to the active connections are returned to the variable changes executed by the Client or Server, etc.</p>
RESOURCES	<p>In this folder there are files relating to the project's Multiple resources (Screens, Accelerators, Menus, Parameterization File and Basic Scripts). Movicon will create a file with the resource's name and extension, based on the resource type, for each resource inserted into the project:</p> <p><i>Screens = .movscr</i> <i>Accelerators = .movacc</i> <i>Menu = .movmenu</i> <i>Basic Script = movbas</i> <b>Parameter Files = movpar</b></p> <p>In addition to this, the same tree structure created also in the 'Project Explorer' window in the Multiple Resource Groups will be reproduced in this folder and therefore with the eventual resources' folder groups.</p> <p>The files containing the communication driver settings to be enables will also be kept in this folder. These files will have the name:</p> <p><b><i>DriverName.drivsettings:</i></b>      <i>file</i>      <i>containing</i>      <i>the</i></p>



	<p><i>Communication Driver settings</i></p> <p><b>DriverName.dynsettings:</b> file containing the list of the Communication Driver's dynamic tasks</p> <p><b>DriverName.dyndrv:</b> file containing the list of tags created dynamically in the variables</p> <p>The Trend configuration files will also be kept in this folder. These files will have the following extension:</p> <p><b>.TSXML</b></p>
--	--



The Project Folders listed above are those which Movicon uses for default. However, they can be customized, creating new ones and with different paths, by means of using the "Project Path Settings".

## Project Backup

Movicon keeps a project build count save in the ".movprj" file's <ProjectBuild> xml tag. When the project's "Increase Project Build Number" execution property is enabled the project's internal build number will increase by 1 each time the project is saved, When this property is not enabled, the build number will remain always the same. When the project's "Save Backup" execution property is enabled, a backup of all the project's files and resources will be done in the ".zip" file in the name of <Project Name>\_<Project Build> each time the project is saved. Therefore, with the project's "Increase Project Build Number" execution property is enabled as well, a new ".zip" file will be created at each project save. When not enabled, the ".zip" file will remain with the same name and overwritten instead each time the project is saved. The folder containing the project will also contain all the backup files of the saves taken place. The ".zip" format has been used so as not to occupy too much space on disk and because it contains files with paths that allow easy extraction.

## Renaming Resources

Movicon has a functionality which consents to automatic refreshing of all references to resources when renamed. The resources concerned are:

- Screens
- Basic Scripts
- Menüs
- Shortcuts

When you rename a resource you do not need to replace the old name with the new one in all the various parts of the project it is used.



**The automatic resource name replacement mechanism only works if the project's execution "Enable Renaming Manager" property is enabled. If not, the aliasing mechanism, described below, will not be used and old names of any renamed resources will have to be replaced manually in the project.**

Movicon uses an "aliasing" mechanism which keeps trace of the resource's original name and its new one. When the resource is remained may times, only its original name and the last name will be traced. This information, the relationship between the resource's original name and its new one, is kept in the ".movprj" file. The tags which enclose this information are different for each resource and are <RenamedScreens> for Screens, <RenamedScripts> for Basic Scripts, <RenamedMenus> for Menus and <RenamedAccelerators> for Shortcuts.

```

...
<RenamedScreens>
<item key="Screen4" value="Main"/>
</RenamedScreens>

<RenamedScripts>
<item key="Basic Script1" value="Basic Script2"/>
</RenamedScripts>

<RenamedMenus>
<item key="Menu1" value="Main"/>
</RenamedMenus>

<RenamedAccelerators>
<item key="Shortcut1" value="Exit"/>
</RenamedAccelerators>
...

```

Where:

**item key:** represents the resource's original name  
**value:** represents the last name used for renaming the resource

When the Renaming Manager has been enabled, it would be a good rule of thumb to re-use original resource names for creating new resources. This kind of operation is always controlled by Movicon displaying this warning message:

Warning! The resource name 'Screen1' has been renamed to 'Screen2' in this project. Do you want to use this resource name? Answering 'Yes' will remove the recorded renamed link.

The above message was generated because the user was trying to create or rename a screen with the 'Screen1' name, which was already used as an original name and then changed to "Screen2". At this point, when answering with 'Yes', the alias link will be deleted and the automatic aliasing mechanism will be annulled for this screen. For instance, if you associated the VAR00001 to a display object and then rename it with VAR00002, the display will result associated to VAR00002. In addition, only one variable, the VAR00002, will be in the RealTimeDB. However, if you insert a new variable with the name of VAR00001 into the database and confirm the use of this name, the alias link will be deleted and the display will result associated to VAR00001 again, which however is the new variable. In this case both the VAR00001 and VAR00002 will result as being in the RealTimeDB as two separate variables.

Lets take another example, if you associate a command to open Screen1 to a button object and then remain the Screen1 to Screen2, the button will result associated to the command to open Screen2. In addition to this, only one screen, screen2, will be present in the Project Explorer window. However, if you then insert a new screen in the name of Screen1, and confirm its use, the alias link will be deleted and the button will result associated again to the open command of Screen1, which is the new screen. In this case both Screen1 and Screen2 will be present in the project and will result as being two separate screens.

When inserting a new resource, Movicon will propose a name which has not yet been used or in use in the project and not been used as an original name for a renamed resource. To make things clearer, if you insert 'Screen1' which then gets renamed to 'Screen2', if you insert a new screen after this, Movicon will propose 'Screen3' as its name.



**WARNING! the "aliasing" mechanism can be deleted by deleting the xml tags which show the list of renamed resources from the ".movprj" file. By doing this, the resources which were renamed will remain with their last associated name, while the references inserted in commands will keep links to the resources' original names. This means that the references to these resources may no longer be valid.**

Automatic replacing of renamed resources is done in different ways according to the object or resource in question, as described below:

**Screen Resources:**

- **Startup Screen Execution Properties:** the new screen name will automatically appear in the project's "Startup Screen" properties
- **Command List Screen Command:** the new screen name will automatically appear in the command's "Screen" property
- **Basic Script:** the original screen name will always appear in the basic script code, for all functions which have a screen name parameter, but the code will be executed in relation to the new screen name

**Basic Script Resource:**

- **Start/Stop Script Execution Property:** the new basic script name will automatically appear in the project's "Start Script" and "Stop Script"
- **Command List Script Commands:** the new name of the basic script will automatically appear in the command's "Script" property
- **Basic Script:** the original name of the basic script will always appear in the basic script code for all function which have a script name as parameter, but the code will be executed in relation to the new basic script name

**Menu Resource:**

- **Command list Menu command:** the new name of the menu will automatically appear in the command's "Menu" property
- **Basic Script:** the original menu name will always show in the basic script code for all functions with a menu name parameter, but the code will be executed in relation to the new menu

The "screen name - menu name" or "screen name - accelerator name" correspondence will not benefit from the automatic name replacement, which means that if you change the name of the screen you will also have to change the name of the menu or accelerator so that they continued being associated to that screen. For example, lets suppose you need to define a "LayOut" screen, a "LayOut" menu and a "LayOut" accelerator in such a way that the screen is loaded, the corresponding menu and accelerator are also activated. Then after having done this, you rename the screen to "PlantLayOut", you will also have to rename the corresponding menu and accelerator with this new name if you want to keep them associated. This also applies to other menus or accelerators associated to the Startup Screen and used for default by all those screens which have not got their own. Also in this case, menus or accelerators must be renamed with the same name of the Startup Screen.

## Project Settings

---

Each Movicon project can be configured to adopt the specific functionalities needed for your customizing requirements.

Each new project has standard configurations therefore the fundamental requirements for being used normally on any machine with acceptable performances for standard applications.

This is usually the programmer's job who carries out the necessary customizing operations to fit the system to the requirements proposed.

The configurations of the system and of the project can be done by the programmer whenever necessary by using the '**Property Windows**'.



You may find it necessary to configure the system before programming the project for which we advise you to check through your own requirements before consulting the possible settings offered by the system.

## General Project Settings

---

By using the General properties of a project you can set which project format will be saved and whether to crypt it or not.

To modify the Project's General Properties, select the object with the mouse and use the Movicon **'Properties Window'**.

### **Crypted Project**

Movicon saves all the project files in XML format to also enable the programmer to open and carry out any changes to the project files and resources with a simple test editor (or even simpler with a XML editor). However, when this property is enabled the project file will be saved in encrypted format and therefore can only be opened with a Movicon editor and no other.

### **Crypted Project Resources**

Movicon saves all the project files in XML format to also enable the programmer to open and carry out any changes to the project files and resources with a simple test editor (or even simpler with a XML editor). However, when this property is enabled the project's resource files will be saved in encrypted format and therefore can only be opened with a Movicon editor and no other.

### **Unicode Project**

When enabling this property all of the project will be saved in UTF-16 Unicode format.

### **Zippped Project**

When enabling this property all of the project will be saved in compressed format. The compression or decompression of files is managed by Movicon automatically in real-time therefore transparent to the programmer. This will only slightly effect performances in the programming and execution phase but has the advantage of occupying less disk space.

### **Save Backup**

When this property is enabled a "Project Backup" file will be created every time the project is saved. For further information on "Project Backup" files please see the section on "Project Files".

### **Increase Project Build Number**

By enabling this property the project build will be increased each time the project is saved. Otherwise, if not enabled, the build will always remain the same and the Backup file, created after each save, will always have the same name and overwrite the previous one. For further information on 'Project Backup Files' please refer to "Project Files".

### **Default Screen Width**

This property sets the default width in pixels with which the Screen window is to be created.

### **Default Screen Height**

This property sets the default height in pixels with which the Screen window is to be created.

### **Default Screen Color**

This property sets the default background color with which the Screen window is to be created..

For further information please refer to the paragraph on "Colour Threshold Settings in Drawings and Controls".

### **Screen Color Number**

This property allows you to select the screen's default set of colors when created.

### **Get Connected Device Screen Size**

This function allows you to set the default screen sizes according to the characteristics of the WinCE target PC where the project is to be exported. Before carrying out this function you need to setup a connection between the desktop PC and the target PC with ActiveSync.

**Alphanumeric Pad Screen**

The name of the Screen to be used as a Alphanumeric Pad instead of the one proposed by Movicon for default is entered in this box. For further information on how customized Alphanumeric Pads work see the section on "Numeric and Alphanumeric Screens".

**Numeric Pad Screen**

The name of the Screen to be used as a Numeric Pad instead of the one proposed by Movicon for default is entered in this box. For further information on how customized Numeric Pads work see the section on "Numeric and Alphanumeric Screens".

**Get Password Screen**

The name of the Screen to be used as the password dialog window for user logins, to replace the one proposed by Movicon for default, is defined here. For further information on how custom Password Dialog windows work please refer to "Custom Password Dialog Windows".

**Expired Password Screen**

The name of the Screen to be used as the expired user password is defined in this box to replace the one proposed by Movicon for default. For further information on how custom expired password screen work please refer to "Custom Password Dialog Windows".

## Project Platform Settings

---

By using the project's Platform properties you can set which platform (Operating System) is to be used for creating the project. To modify the project's Platform, select the object with the mouse and then use the Movicon **'Properties Window'**.

**Windows XP/Vista**

This property enables project development for Windows XP/Vista platforms.

**Windows CE**

This property enables the project development for Windows CE platforms.

**Client WinXP/Vista**

This property enables project development for Windows XP/Vista Client platforms.

**Client WinCE**

This property enables the project development for Windows CE Client platforms.

**WebClient Applet**

This property enables the project development for J2SE Client (Java to Standard Edition, such as Windows, Linux or other Java-enabled platforms).

**WebClient Middlet**

This property enables the project development for J2ME Client (Java to Machine Edition, such as JavaPhones or mobile phone platforms).

## Project Path Settings

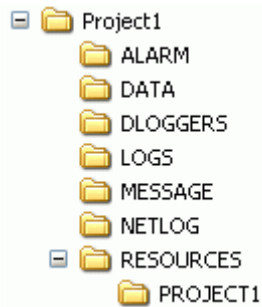
---

The project files can be organized in secondary folders according to the default settings or those carried out in the Project's **'Working folder Settings'**.

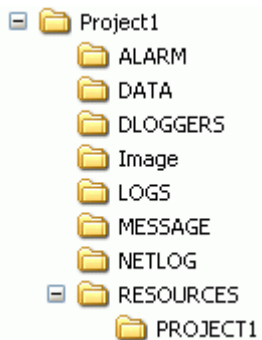
To modify the project Paths properties, select the object with the mouse and then use the Movicon **'Properties Window'**.

Normally the project files are organized according to a folder structure in which the data is organized in a logical order. At startup Movicon creates a few default folders from the main project folder where the system will organize its data.

The structure of the working folders can be customized as pleased, by declaring in the project's **'Path'** properties how you wish to organize your files and those of the system's.



An example of how standard Movicon project working sub-folders are organized on disk.



An example of how customized working sub-folders are organized with the 'Image' folder where all the project's images will be inserted.

#### Alarms Path

The working folder, in which Movicon will file any comments associated to alarms by the operator, can be declared or selected in this box. The 'Alarms Comment' editing operations are described in the relative chapter on project Alarms. The selection of an existing folder can be done with the '...' button on the right.

If the folder does not already exist it will be created by the system upon project startup.

#### Data Path

The working folder can be declared or selected in this box in which Movicon will file data relating to the retentive variables or the "Watch Window" settings. You can use the "..." button on the right to select an already existing folder. If a folder does not exist one will be created at the project startup.

#### Images Path

The working folder, in which Movicon will search for the bitmap or jpg type images used in the project, can be declared or selected in this box. The selection of an already existing folder can be done with the '...' button on the right.

If the folder does not exist it will be created by the system upon project startup.



Starting from the 956 build Movicon will use the predefined "IMAGES" folder which will be created as the project folder when this property is not used.

- Images set as screen backgrounds will first be searched for in the folder preset for containing images, then in the same folder in which the screen resource was saved, and then in the project folder.
- Images set in controls, alarms or in variables will first be searched for in the folder preset for containing these images and then in the project folder.

In addition to this, when an image is selected its copy will be saved in the

preset folder containing images ('IMAGES' for default). This allows you to keep all your images localized in the project folder so you don't have problems localizing images when running the project in other computers or when exporting it to WinCE devices.

#### **Resource Path**

The working folder in which Movicon is to save the multiple project resources (Screens, Menus, Accelerators, etc.) is declared or selected in this box.

You can use the "... " button on the right to select existing folders.

If the folder does not exist it will be created by the system upon project startup.

#### **Logs Path**

The working folder in which Movicon is to file recorded data from the project's Historical Log Events and Variable Trace is declared or selected in this box. You can get further information on these functions from the 'Historical Log' chapter in the Manual.

You can use the "... " button on the right to select existing folders.

If the folder does not exist it will be created by the system upon project startup.

#### **Data Loggers Path**

The working folder in which Movicon is to file recorded from the project's Data Loggers is declared or selected in this box. You can get further information on this function from the 'Data Loggers' chapter in the Manual.

You can use the "... " button on the right to select existing folders.

If the folder does not exist it will be created by the system upon project startup.

#### **Network User Path**

This box is used for declaring or selecting the work folder in which Movicon will insert the Log files of the network connection coming from any eventual clients. If this field is left empty, Movicon will use the project's "NETLOG" folder for default.

You can use the "... " button on the right to select existing folders. If the folder does not exist it will be created by the system upon project startup.

#### **HTML Help File**

A help file to be associated to the project in HTML help format can be selected in this box.

The Topic of this Help file can then be viewed through the **"Help Command"** which can be associated to the Movicon controls.



When inserting the Help file in the project's **".\Resources\ProjectName"** folder, Movicon will be able to use the path relating to the file as to avoid using any absolute paths due to moving the file to other PCs or to different folders.

## **Project Execution settings**

---

By using the Execution properties of a project you can set by enabling or not enabling the procedures for the project's statistical data and any commands or basic routines to be executed upon the project startup or shutdown.

To edit the project's Execution properties, select the object with the mouse and use the Movicon **'Properties Window'**.

#### **Enable Statistics**

This property is used for enabling or disabling project statistical data acquisition. This data can be seen using the "Watch window".

#### **Enable Admin Alerts**

This property is used for enabling or disabling the sending of 'Admin Alert' messages to a PC Server.

#### **Admin Alerts Server**

This name of the Server where the 'Admin Alert' messages are to be sent is entered in this box.

### Startup Screen

The name of the Screen to be displayed at project startup is entered in this box. When this property is not set, therefore without any Startup Screen specified, Movicon will check to see if one exists in the name of 'Main' at the startup of the project's runtime and will load it if it does. However, when this property has not been set with a specific startup screen and a 'Main' screen does not exist, Movicon will randomly load one of those present in the project at startup.

### Startup Script

The name of the Basic Script to be executed upon project startup is entered in this box. The selection of the Script can also be done by using the '...' button on the right.

### Shutdown Script

The name of the Basic Script to be executed upon project shutdown is entered in this box. The selection of the Script can also be done by using the '...' button on the right.

### Startup Commands

The Movicon '**Command List**' is opened by using this button through which you can set a list of one or more commands which must be executed upon project startup. For further information on the commands available please refer to the paragraph on "**Command List**".

### Shutdown Commands

The Movicon '**Command List**' is opened by using this button through which you can set to list of one or more commands which must be executed upon project shutdown. For further information on the commands available please refer to the paragraph on "**Command List**".

### Pre-load Screens

This option allows all those screens marked with "**Not destroyable**" option to be preloaded at the project startup. This option is not set for default when new screens are created in the project.

When this option is set, Movicon will scan all existing screens at the project run startup and preload in memory all those which are marked with the "**Not Destroyable**" option.

This operation, inserted to start from the 956 build, allows maximum velocity when changing pages - when needed - due to the fact that the screen to be opened will have been already preloaded before being activated.



Note: This operation also happens for **child project** screens only when the child project is marked as executable and in auto run, and when the screens are marked with the preload option.

### Start Full Screen

Enabling this option will startup the project in Full Screen mode, therefore the Screen window will be displayed without the title bar.

### Show Status Bar

Enabling this option will also display the Movicon status bar during the Runtime phase.

### Show Output Window

Enabling this option will also display the Movicon Trace window during Runtime. The Trace, Output or Debug window is very handy to have especially during the project Debug phase.

### Enable Renaming Manager

This option allows to enable or disable the variables and resources Renaming Manager (see "Renaming variables" and "Renaming Resources"). When this property is enabled, renaming a variable or a resource does not require to replace the previous name with the new name in the project, where the variable or the resource name has been used. Disabling this property, the automatic replacement of variables or resources name will not take place.

### Hide Desktop

When enabling this option the Desktop's icons will be hidden and therefore inactive. Please note that the Windows "Show Desktop" button presented in the 'Quick Start' of "Task Bar", permits the Desktop to be reactivated. To avoid this from happening it is necessary to disable the 'Task Bar' as well.

### Hide Start Button

When enabling this option the Windows Start button will be hidden and therefore inactive.



**Hide Task Bar**

When enabling this option the 'Task Bar' will be hidden and therefore inactive.

**Hide System Clock**

When enabling this option the clock, presented in the Windows' 'Task Bar', will be hidden.

**Disable Task Switching Keys**

When enabling this option the Windows switching keys such as CTL+EXIT, ALT+TAB, etc, will be deactivated.

**Disable Task Manager**

When enabling this option the Windows' Task Manager will be deactivated.

**Disable Ctrl+Alt+Del**

When enabling this option the use of the Windows Ctrl+Alt+Del will be deactivated.

**ROT**

When this option is enabled the "PmeDocCmdTarget" interface will be recorded in the Windows' ROT (Running Object Table) each time a new project is opened for the first time. This interface is available to any script engine which supports OLE Automation (such as Visual Basic, by means of using the GetObject function). This allows you to keep the currently used "PmeDocCmdTarget" object so that you don't have to create a new one. By accessing the methods and properties of this interface you will also be able to access the project's variables in order to read or write them. The syntax for creating the object from Visual Basic code is as follows:

```
Dim Movicon As Object
Private Sub Form_Load()
    Set Movicon = GetObject(, "Movicon.Document.11.0")
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Set Movicon = Nothing
End Sub
```

**ROT Any Client**

When this option is enabled the "PmeDocCmdTarget" interface will be recorded in the Windows' ROT (Running Object Table) in non safe mode. This allows you to use the "PmeDocCmdTarget" also when the project is started as Windows service.

**Show Service Sys Tray Icon**

If the project is executed as "Windows Service", this property allows you to decide if show or hide the icon of the service on the Windows' Application Bar.

**Show Service at Log On**

If the project is executed as "Windows Service", this property allows you to decide if open and display the project screen (Movicon application) of the started service, **automatically after the Log On for the operating system.**

## Heap Manager Settings

---

Movicon allows you to get optimized memory allocation management during project runtime. Each project object group can be set with a quantity of memory that Movicon will allocate in fixed mode. If this management is not enabled, Movicon will allocate and deallocate memory as required during the Runtime phase. However, this may cause fragmentation in the Ram with consequent undesired memory consumption, especially when the project remains in run mode for a long time. On the other hand, when this management is enabled, Movicon will allocate a fixed portion of memory for each group of objects and will use only that fixed portion. These memory portions are called "Heaps". Caution must naturally be taken to set the right values to avoid errors of insufficient quantity of allocated memory during the runtime phase. It must also be taken into consideration that by enabling this management the Ram allocation may turn out to be higher than expected when not enabled. This is due to the fact that Movicon allocates all the configured memory even when not all of it is to be used.



The Heap Management is always remains disabled when the project is run from the development environment. However it is managed for the Movicon RunTime .exe, Movicon Service .exe, Movicon CE .exe and Movicon .exe modules only when run with the /R. option.

The Heap values can be set manually or with the appropriate button, which does nothing other than get the maximum values requested during the last project run. The heap values are kept traced during runtime (the current value and the maximum values reached) for each group of objects. These values can be viewed through the Watch window (the maximum values reached are those shown in brackets).

Item	Value
- demo11	
# Heap Manager : Alarm Scripts	0 (0)
# Heap Manager : Draw Scripts	0 (0)
# Heap Manager : OPC Items	0 (0)
# Heap Manager : OPC Groups	0 (0)
# Heap Manager : OPC Servers	0 (0)
# Heap Manager : Alarms	31 (31)
# Heap Manager : Variables	747 (771)
# Heap Manager : Draw Dynamics	51 (253)
# Heap Manager : Draw GridRow	0 (0)

Therefore, once the project has been put into run mode you can return to development mode and by using the "set values from last run" all the heap manager properties will be updated with the values of the last run.

The Heap values can also be set by using the appropriate registry keys described in the "List of Movicon Registry Keys" section. However, the registry values are by-passed by the project values when the heap manager is activated.

#### Set Values from the last Run

This command allows you set the Heap values for each group based on the last value obtained during the Runtime phase.

#### Memory Used with the current Heap Values

This field shows the total value in memory Kbytes which will be allocated based on the Heap values set.

#### Recalculate memory Used

This command updates the value shown in the "Memory Used with the current Heap values" property with the values set at that moment.

#### Enable Heap Values

This option box is used for enabling or disabling the memory through the Heap Values. When disabled, any Heap registry key inserted in the Windows Registry will be bypassed by the values set in the project.

#### Heap...

These are used for inserting Heap Values for each group of objects. Keep into account that the values are not expressed in either KBytes or as object numbers, but is the heap quality necessary for managing that type of object. The meaning for each value is:

**Rectangle Heap:** maximum heap size for "Rectangle" object allocations (0 = unlimited)

**Alarm Window Heap:** maximum heap size for "Alarm Window" object allocations (0 = unlimited)

**Historical Log Window Heap:** maximum heap size for "Historical Log Window" object allocations (0 = unlimited)

**DataLogger-Recipe Window Heap:** maximum heap size for "Historical Log Window" object allocations (0 = unlimited)

**TraceDB Window Heap:** maximum heap size for "TraceDB Window" object allocations (0 = unlimited)

**Button Heap:** maximum heap size for "Button" object allocations (0 = unlimited)

**Chart Heap:** maximum heap size for "Chart" object allocations (0 = unlimited)

**Polygon, Multiline, etc heap:** maximum heap size for "Polygon, Multiline, etc" object allocations (0 = unlimited)

**OLE object Heap:** maximum heap size for "OLE" object allocations (0 = unlimited)

**OCX/ActiveX Object Heap:** maximum heap size for "OCX/ActiveX" object allocations (0 = unlimited)

**Group Object Heap:** maximum heap size for "Group Box" object allocations (0 = unlimited)

**Embedded Screen Heap:** maximum heap size for "Embedded Screen" object allocations (0 = unlimited)

**Trend Heap:** maximum heap size for "Trend/Data Analysis" object allocations (0 = unlimited)

**Meter Heap:** maximum heap size for "Meter" object allocations (0 = unlimited)

**Display Heap:** maximum heap size for "Display" object allocations (0 = unlimited)

**Tab Group Heap:** maximum heap size for "Tab Group" object allocations (0 = unlimited)

**Scheduler Window Heap:** maximum heap size for "Scheduler Window" object allocations (0 = unlimited)

**List Box Object Heap:** maximum heap size for "List Box/Combo Box" object allocations (0 = unlimited)

**Grid Heap:** maximum heap size for "Grid" object allocations (0 = unlimited)

**Dynamic Object Heap:** maximum heap size for allocations of those objects set with "Animation" properties (0 = unlimited)

**Tag Heap:** maximum heap size for project Tag allocations (0 = unlimited)

**Alarm Heap:** maximum heap size for "Alarm" object allocations (0 = unlimited)

**OPC Item Heap:** maximum heap size for "OPC Item" object allocations (0 = unlimited)

**OPC Group Heap:** maximum heap size for "OPC Group" object allocations (0 = unlimited)

**OPC Server Heap:** maximum heap size for "OPC Server" object allocations (0 = unlimited)

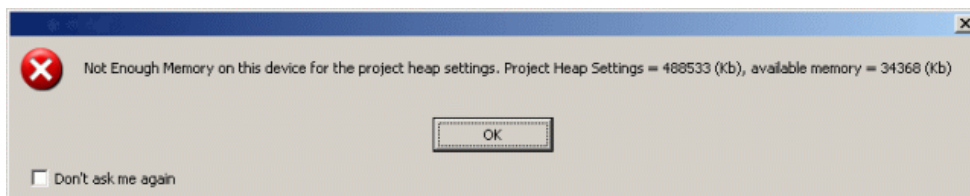
**Drawing Script Heap:** maximum heap size for allocation of script code associated to "Drawings" (0 = unlimited)

**Alarm Script Heap:** maximum heap size for allocation of script code associated to "Alarms" (0 = unlimited)

The "0 = unlimited" value means that no fixed heap portions are allocated, but will be allocated dynamically as needed.

## Upload projects to WinCE

Movicon controls that the heap settings do not exceed the maximum amount of memory installed on the device (indications are given in approximate value counts which are not precise to the byte) when uploading projects to the WinCE device (ActiveSync only). The Movicon CE runtime also carries out this control before starting with these values and gives a warning message when there is insufficient memory:



The heaps in WinCE are allocated to the global memory (memory available for the programs) and do not occupy memory in the Movicon CE .exe process. In this way the fixed 32Mbytes memory, which each WinCE process can use, is by-passed by these Heap managers. Therefore you can check each screen as not destroyable providing that there is enough memory on the device.

## Project Historical Log Settings

---

Movicon manages the Historical Log of events through a completely configurable registration engine. Movicon automatically records on Log all the significant system events, events inherent to communication drivers and plant alarms and messages which will only be recorded if the programmer has enabled this property in each single alarm.

The Log recording modalities for Movicon projects can be set through the Movicon **'Properties Window'** after having clicked the project name desired (from the 'Project Window').

The Historical Log's data recordings are done on DataBase files by exploiting the operating system's ODBC links or on text files using the Movicon IMDB manager.

### **Enable Log Recording**

This property allows you to enable the Historical Log. Unmarking this property will disable the Historical Log database recordings, whether using the IMDB or ODBC manager.

However the Historical Log will continue to record in text files within the project's "LOGS" folder. This new property is also enabled for default to guarantee compatibility with previous versions.

### **Use IMDB Manager**

See paragraph **"IMDB Settings for Recording data"** in the section on **"IMDB"**.

### **Shared Tables**

See paragraph **"IMDB Settings for Recording data"** in the section on **"IMDB"**.

### **Save XML File**

See paragraph **"IMDB Settings for Recording data"** in the section on **"IMDB"**.

### **Crypt File**

See paragraph **"IMDB Settings for Recording data"** in the section on **"IMDB"**.

### **Write Behind Delay**

See paragraph **"IMDB Settings for Recording data"** in the section on **"IMDB"**.

### **Max Numero Rcor ds**

See the **"IMDB Settings for Recording data"** paragraph in the **"IMDB"** section.

### **Keep the DB Connection open**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

### **Send Admin.Alerts**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

### **Max. Error Number**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

### **Max. Transactions**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

### **Max. Cache Size**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

### **Max. VarChar Precision**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

**ODBC DSN**

See paragraph "DBMS Settings for Recording data on DataBase" in the section on "Historical Log Management".

**ODBC DSN User**

See paragraph "DBMS Settings for Recording data on DataBase" in the section on "Historical Log Management".

**Time Column**

This setting permits you to insert the name of the Historical Log Table's Time Column. When left blank the default name will be used instead. The Time column indicates the recording's date and time in GMT (Greenwich Mean Time). GMT is universally used for time reference. Time zones are to be calculated by starting from GMT 00. hrs.

**Local Time Column**

This setting permits you to insert the name of the Historical Log Table's Local Time Column. When left blank the default name will be used instead. The Local Time Column indicates the recording's date and local time.

**MSec Column**

This setting permits you to insert the name of the Historical Log Table's MSec Column. When left blank the default name will be used instead. The MSec Column indicates the milliseconds relative to the recording's time.

**User Column**

This setting permits you to insert the name of the Historical Log Table's User Column. When left blank the default name will be used instead. The User Column indicates the name of the user active the moment the recording took place.

**Event Type Column**

This setting permits you to insert the name of the Historical Log Table's Event Column. When left blank the default name will be used instead. The Event Column indicates the event type recorded (ie. Alarm ON, Alarm OFF, System, etc.).

**Num. Event Type Column**

This setting permits you to insert the name of the Historical Log Table's Event Number Column. When left blank the default name will be used instead. The Event Number Column reports the recorded event identification number.

**Desc. Column**

This setting permits you to insert the name of the Historical Log Table's Description Column. When left blank the default name will be used instead. The Description Column reports the recorded event description.

**Sub Event Column**

This setting permits you to insert the name of the Historical Log Table's Sub Event Column. When left blank the default name will be used instead. The Sub Event Column reports different information for each table. In the Alarms Table for example, this field reports the Alarm's text.

**Comment Column**

This setting permits you to insert the name of the Historical Log Table's Sub Event Column. When left blank the default name will be used instead. The Comment Column reports the alarm's duration for the Alarm Table.

**Duration Column**

This setting permits you to insert the name of the Historical Log Table's Duration Column. When left blank the default name will be used instead. The Duration Column reports how long the event in question lasted.

**Unique ID Column**

This setting permits you to insert the name of the Unique ID Column in the Historical Log tables. The default name will be used when left blank. This column shows the alarm's Unique ID value when dealing with alarm tables. However, this field remains blank for the "System" and "Drivers" tables.

**Transaction ID Column**

This setting permits you to insert the name of the Transaction ID Column of the Historical Log tables. The default name will be used when left blank. This column shows the alarm's Transaction ID value when dealing with alarm tables. However, this field remains blank for the "System" and "Drivers" tables.

**Alarms Table**

This setting lets you insert the name of the Historical Log table which will contain the messages inherent to project's 'Alarms'.

**Drivers Table**

This setting lets you insert the name of the Historical Log table which will contain the messages inherent to the project's 'Drivers'.

**System Table**

This setting lets you insert the name of the Historical Log table which will contain the messages inherent to project's 'System Messages'.

**Alarm Max.Age**

This property lets you set how long the Alarm messages are to be kept internal the database. The Default setting is 180 days but can be changed according to what is required, keeping in mind how many recordings will be done in the time interval specified.

**Driver Max.Age**

This property lets you set how long the Communication Driver messages are to be kept internal the database. The Default setting is 180 days but can be changed according to what is required, keeping in mind how many recordings will be done in the Time interval specified.

**System Max.Age**

This property lets you set how long the System messages are to be kept internal the database. The Default setting is 180 days but can be changed according to what is required, keeping in mind how many recordings will be done in the Time interval specified.

**Create DB Table**

This command executes the creation of the Historical Log tables internal the database. If tables already exist they will be cancelled by being recreated with the new ones upon the execution of this command. This means that any data recorded beforehand will be lost.

## Child Project Options Settings

---

By using the Child Project Options properties you can set the start modalities of any declared Child Projects.

To modify the Child Project Options Properties, select the project from the 'Child Project List' group from the Project Explorer window with the mouse and then use Movicon **'Properties Window'**.

**Startable**

When this property is enabled the child project can then be run. This means that all the sever part of the project, being the Drivers, OPC, Networking etc, will be started. When a project is not enabled to start you can nevertheless access the child screens from the parent project.

**Auto Startup**

This property permits the child project to run automatically when the parent project is put into execution. In order for this setting to take effect you must also select the 'Startable' property.

**Network Server**

This property permits you to set the Networking Server Project's name from which the child project is to update the values of its variables.

### Child Project Name

This property permits you to associate a name to the child project. When this field is left empty the child project's name will be used instead.

## Project Spooler Manager Settings

---

The "Spooler Manager" manages the sending of project events to a LPT or COM port. The events managed by the Spooler are: Alarm ON, Alarm ACK, Alarm OFF, System. Comm.Driver. The use of LPT port needs the "printer spooler" service which is active in Windows. This service is normally always active. The "Spooler Manager" needs "data.spool" xml files (created in the project's "DATA" sub-folder) for saving certain information such as the current line number and the spooler's buffer.

To change the project's "Spooler Manager" properties, select the name of the project with the mouse and use the Movicon **"Properties Window"**.



**The spooling management for printing events directly does not support texts with Unicode code.**



**Cases where the printer in use does not support ASCII code for formatting text (ie. as with many inkjet printers), you will need to set the print styles of each event with the "10 cpi" character option and not set any of the style options.**

### Printer Port

The Port to be used for sending texts to be printed. LPT ports from 1 to 3 and COM ports from 1 to 9 are supported.

Where an opening of a COM port is concerned, you can set the port's opening parameters by using the following format:

COMx:BAUD=9600 PARITY=N DATA=8 STOP=1

### Read Interval Timeout

This value is used after the selected port has been opened for setting a timeout on the two character reception (IOCTL\_SERIAL\_SET\_TIMEOUTS). You may find it useful to modify this value if the serial printers or printers being used are very slow.

### Read Timeout

This value is used after the selected port has been opened for setting a timeout on the reception (IOCTL\_SERIAL\_SET\_TIMEOUTS). The default value is 10000 milliseconds.

### Write Timeout

This value is used after the selected port has been opened for setting a time on the transmission (IOCTL\_SERIAL\_SET\_TIMEOUTS). The default value is 10000 milliseconds.

### # Lines per Page

This value allows you to decide how many printed lines are needed before sending the page end to the port. To print a continuous page you should enter the "0" value so that no change page takes place. When using single sheets of paper you should enter the number of line per sheet so that a change page takes place on the last line printed.



Events to be printed are sent immediately to the selected port. Laser or ink jet printers must be setup beforehand so their buffer contents are printed only when the end page character is received.

### Max. Buffer Size

This property determines the buffer memory size in Kbytes used for the spooler. When the buffer is full the oldest events will be deleted from the buffer in function with the most recent ones and will not be printed. A message will be traced in the system log for each event deleted from the Spooler Buffer:

"The Spooler Buffer has reached the maximum value. Entry '<event text>' has been removed"

The buffer memory is saved in the "data.spool" file so that those events not printed due to unforeseen application aborts can be printed.

#### Time From

The spooler can be made active for a specific time interval. The events which are generated outside this time interval will not be printed. This property is used for entering the start time in hours and minutes.

#### Time To

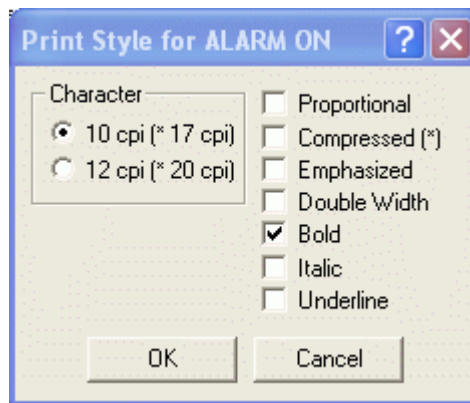
The spooler can be made active for a specific time interval. The events which are generated outside this time interval will not be printed. This property is used for entering the finish time in hours and minutes.

#### Enable Spooler...

You can activate or deactivate the spooler for each event category (Alarm ON, Alarm ACK, etc). The spooler can also be activated or deactivated for each alarm threshold ( "Print" style properties) and for each Basic Script ( "Spooler" Script Debug Output properties ).

#### Event Style...

The message print style can be set for each event category. The following dialog window will open by clicking on the appropriate command button of each event (...) through which the print characters can be set:



The meaning of these settings are as follows:

- 10 cpi or 12 cpi: print character size (small or big)
- Proportional: the text is printed in proportion to the characters
- Compressed: the text is printed in compressed mode
- Emphasized: the text is printed emphasized
- Double Width: the text is printed with doubled width
- Bold: the text is printed in bold
- Italic: the text is printed in italic
- Underline: the text is printed underlined



**The direct event's printout (spooling) doesn't support text using Unicode.**



**In case the printer doesn't support ASCII codes for text formatting (a lot of inkjet printers, for example) you have to set properly the print style properties and, for each event, the character style option at "10 cpi" and don't set no other style options.**

## tools

Movicon uses some tools, developed by Progea as usual, to carry out certain operations or functions. Below you will find a list of these tools with a brief description on what they do. For further information on how they work and any application configurations please consult the Help online of the related application.



**Upload/Download service**

The executable file "TCPUUploadServer.exe" is installed with Movicon setup and allows to use, for a desktop platform too, the service for projects upload/download to/from a remote client. For further information on this subject please see "Projects Upload/Download " in Movicon CE user manual.

**Alarm Dispatcher**

Movicon uses this application, "Dispatcher.exe", to configure the SMS, Voice, Fax alarms notifying.

This tool can be also used within basic script routines for sending messages that have nothing to do with the alarms.

**CSV Tag Importer-Exporter**

This application, "CSVTagIE.exe", allows the importing/exporting from the Real Time DB in csv. files.

The import and export commands function on all the data from the project's RealTimeDB: variables, structure prototypes, installed communication drivers (the settings of each communication driver are not managed by the CSVTagIE because they are saved on external files outside the ".movrealtimedb").

**Importer Command:** the CSVTagIE updates the destination RealTimeDB without rewriting it completely. This allows variables to be added without eliminating everything already inserted with in the internal. Upon arriving at the first already existing variable a request will be made to overwrite all the other existing variable; if "No" is pressed, none of the existing variables will be overwritten. If "Cancel" is pressed, a request will be made to overwrite each existing variable one by one.

**Exporter Command:** the csv. file is always overwritten completely (if it already exists, a window will appear asking for confirmation). Each line exported in the .csv file starts with a character which identifies the data type as described below:

V = Variable  
S = Structure Prototype  
D = Communication Driver

You can also select which column separator to use for exporting the csv. file. The characters you can choose are: ";", ":", "||".

**PrnWnd.exe**

This program, "PrnWnd.exe", is installed in Movicon installation folder and consent the printing of screens or the Window desktop. The raster print is a graphical print mode which creates a uniform bitmap of the page and sends it to the printer. The syntax is:

**PrnWnd.exe [-h<%d>] [-d] [-s] [-t<%s>] [-l]**

where:

**[-h<%d>]:** %d is the handle's decimal value of the window to be printed. If the handle is not specified, all of the desktop will be printed

**[-d]:** dialog window appears for choosing the printer and the print options, otherwise the default printer will be used

**[-s]:** hides the window which indicates the print in action

**[-t<%s>]:** attributes the print document title

**[-l]:** forces a vertical print (portrait)

The tool can be activated using the commands from the "Capture and Print" Commands List of the "Screen", "Run Application" and "System" groups. This tool can also launched directly from the Movicon script code where this command should be used:

**Shell("PrnWnd.exe", vbNormalFocus)**

The tool is available for these following platforms:

Win32  
WinCE (SSDK ARMV4I)  
WinCE (SSDK x86)

### **Tag-Alarm Wizard**

This application, "AlarmGen.exe", allows lists of alarms to be created quickly by using a wizard.

### **CleanLog**

This application, "CleanLog.exe", is used by Movicon for deleting old networking log files exceeding a preset life span. This date is calculated by Movicon according to the project's "Log and Security" property settings from the Networking Services resource.

This application can also be used by the programmer to cancel files. It can be called up by using the command line by passing the following parameters:

```
CleanLog.exe -d<x> [-r] <path>
```

Where:

-x = number of days

-r = also deletes the files in the sub folder

Path = path of the folder where the files are found

CleanLog deletes all the files contained in the folder <path> which are dated older than the actual date + <x> days.

### **ReportViewer**

This application, "ReportViewer.exe", is used by Movicon for displaying Report files created with Crystal Report and associated to DataLoggers, Recipes, Historicals etc.. It can be ran from a command line by passing some parameters to customize settings which otherwise are set by Movicon with default values:

```
ReportViewer.exe [-p] [-t] [-g] <report file name>
```

Where:

-p = sends report to printer without passing through print preview

-t = displays the command toolbar

-g = displays group area

You can run ReportViewer using the "System/Launch Application" command from the Movicon commands list, adding the proper parameters in the "Command Line" property, or using a batch file.

### **ReportViewerNET**

This application, "ReportViewerNET.exe", is used by Movicon for displaying Report files created with Report Designer and associated to DataLoggers, Recipes, Historicals etc.. It can be ran from a command line by passing some parameters to customize settings which otherwise are set by Movicon with default values:

```
ReportViewerNET.exe [-f] [-p] [-q"select * from Alarms"] [-cDSN] [-eX]  
<ReportFileName>
```

where:

-f = allows to open the report with the Report Designer, instead of showing the preview

-p = prints the report without print preview

-q "query" = allows to specify a query string to select data to extract and be shown in the report

-cDSN = where DSN is the name of the ODBC connection to be used for connection to the database (i.e. TestProject\_HisLog). If not specified, the connection set by the Report Designer when the report is first created is used (in this case it can no more be accessed or modified)

-eX = exports data in a different format. X stays for the format, valid values are: 2 (pdf), 3 (html), 4 (txt), 5 (csv), 6 (xls), 7 (Mht), 8 (rtf), 9 (jpg). The export file is created in the folder where the report is placed, with the name of the report itself

<ReportFileName> = path and name of the report file to be shown (i.e. "E:\ProjectPath\TestProject\DLOGGERS\TestReport.repx")

You can run ReportViewerNET using the "System/Launch Application" command from the Movicon commands list, adding the proper parameters in the "Command Line" property, or using a batch file.

#### **Run Time users**

This application, "EditUsr.exe", is used by Movicon to create and modify the Run Time users.

#### **LangManager**

If during Movicon setup more than a language has been chosen in the "Languages" dialog window, you can change the language to be used for Movicon development environment. This application, "LangManagerPlus.exe", can be used to change the language for Movicon development environment. Movicon should be closed before launching it.

Launching the application a dialog window listing the available languages is shown. Once the language selection is confirmed pressing the "OK" button, wait until the hourglass cursor disappears, then restart Movicon: the new language will be active.

#### **Development Code Info**

This application, "RegDevCode.exe", is used to show Movicon development (or registration) code.

## Visual Studio SourceSafe 2005 Integration and Support

---

Visual Studio SourceSafe integration and support consents the development management of Movicon projects to be divided among diverse users. The Microsoft SourceSafe tool has been purposely created to manage project designing in teams and to keep track of all the modifications taken place in the project. This support has also been added to the symbol library so that symbol categories can be shared between the users by always using SourceSafe.



**Full support to project sharing is guaranteed only by using Visual Studio SourceSafe 2005, or later.**

### **Visual Studio SourceSafe 2005**

---

As mentioned above, in order to manage Movicon project sharing you will need to install the Visual Studio SourceSafe 2005. Once installed you will then need to create a reference Database where all the updates on the modifications to the project, by the various users, are to be kept. When creating this database you can choose whether to create a new one or connect to an existing one. If the reference Database for the project already exists you will just need to connect to that database.

For further information on how the Visual Studio SourceSafe 2005 works, please refer to the Microsoft guide.

### **Using SourceSafe in Movicon**

---

When opening a project, Movicon controls whether Visual SourceSafe has been installed, and then integrates its potentiality to the full. All of the SourceSafe management will be activated if a project has already been created in the Visual SourceSafe with the same name of the project opened by Movicon. Movicon will therefore ask if you want to load the last version of the project before opening it. It will then be possible to do the following operations in each of the project's resources: "Get Latest Version", "Check Out for Edit", "Check In..." and "Undo Check Out".



**A problem has arisen when using Windows XP in Italian which does not permit files with names in Unicode chars to be added in SourceSafe. For instance when trying to create in SourceSafe a project with the following Unicode name "Фисв.movprj", the "Incorrect Access Code (wrong parameter)" error is returned.**

when a Movicon project is opened in SourceSafe all the resource icons open with a padlock showing on top. This means that these resources are locked and you will need to use the "Check Out for Edit" command to unlock it for editing if not already being edited by other users.

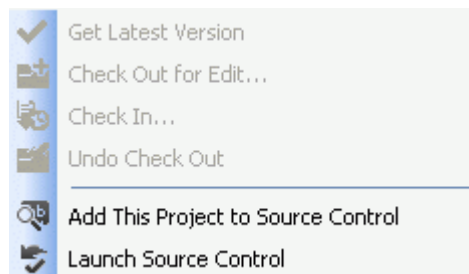


**When a project is shared in SourceSafe, some of the project's options may only be set if the project resources are not being edited by another user. The options are all those which require a complete save of all the resources in the project: "Crypted Project Resources", "Unicode Project" "Compressed Project". In cases when one of these options is modified, Movicon will ask the user to check out all the project's resources from the SourceSafe to then resave them based on the new project options. If this operation does not terminate successfully (ie. because a resource if opened in editing mode by another user), the options will be restored back to their original values.**

## SourceSafe Commands

---

All the command for the SourceSafe management are available from the "File > Source Control" menu or from the "Source Control" menu which appears with a right mouse click on anyone of the project's resources:



### Add This Project to Source Control

This command allows you to create a new project in SourceSafe and automatically adds all the local project's resource files to the project in SourceSafe. In this way the project can be shared with all the other SourceSafe users who may need to modify it. Projects created in SourceSafe can be deleted only by using the Sourcesafe application, which is done by using the "Launch Source Control" command. The project created in SourceSafe will have the same name as the local project's.

### Launch Source control

This command allows you to launch the Visual Source Safe program. Visual Source Safe will open in the last database opened using the user currently using Movicon.

### Get Last Version

This command allows you to read the last version of the resource selected in the Project Explorer window from the project in Visual SourceSafe. The last version of the current resource file in the SourceSafe project will be copied locally, and the user will be asked whether they wish to reload the project to put this command into effect.

### Check Out for Edit...

This command allows you to extract a resource file from the SourceSafe project. In this case the extracted resource will be assigned in the SourceSafe project to the user who has extracted it and no other SourceSafe user will be able to edit it until the that assigned user has checked it back in. Please note that this command does not consent you to get the last resource file version from the "SourceSafe" project. If you wish to get the last version you must use the "Get Last Version" command.

### Check In...

This command allows you to file a resource file in the SourceSafe project. Modifications carried out to the resource will be validated and filed in the sourceSafe project. In this way the resource will again be available for checking out with the "Check Out for Edit..." command by any one of the sourceSafe users. Movicon automatically files all the checked

out resources when the project is closed. This consents the project's resource to be freed automatically for other users.

#### **Undo Check Out**

This command allows you to cancel modifications done to a resource. The local resource file will be restored to its last valid version of the file in the SourceSafe and the user will be asked whether they wish to reload the project to put this command into effect. Please note that the moment the project is reloaded it will no longer be possible to cancel any other modifications in other resources due to the fact that they are automatically validated and filed before the project is reloaded. For instance, if two different resources have already been edited the "Undo Check Out" command may only be executed on one of these two resources as the other one will be automatically validated and saved when the project is reloaded.

The "Undo Check Out" command will not be enabled until the modified resource has been saved.

Movicon cancels the check out automatically and exits from editing the resource without saving the its modifications, automatically freeing it for other users.

#### **Registry Keys**

---

You can modify some of the Windows configuration registry keys that Movicon reads to change certain behaviour due to SourceSafe integration. These registry keys are described in the "SourceSafe Keys" section.

#### **SourceSafe in Symbol Libraries**

---

SourceSafe support is also available for sharing the 'Symbol Libraries' categories. The command are the same ones available when using the project and can be accessed from the menu with the same name which appears with a right mouse click on any one of the library's categories. Therefore you can create a SourceSafe project for the Symbol Libraries containing all the categories inserted and manage the 'Get Latest version', 'Check Out for Edit..' and 'Check In' operations.

When opening the Symbol Libraries, Movicon will verify whether a project relating to the Symbol Library exists in the SourceSafe and if affirmative, will ask the user if they wish to have the latest version of all the files.

## 5. Real Time DB

---

***This section describes how Variables are managed within Movicon projects, memory areas, the Variables List and their functionalities.***

The purpose of a supervision system is to represent or acquire variables from the plant, process them and eventually return them back to the plant whether determined by settings or commands executed by the operator or generated by internal processing. The logic variables, constitute the information (deriving from digital or analogic values) which, combined together, consent Movicon program management.

The project variables, also called Tags, are defined in the Variable List(Tags) contained in the project's Real Time DB resource.

The Movicon realtime database is based on an extremely efficient and optimized technology. The kernel of the platform is based on a innovative system technology, even if based on XML, and is managed in runtime in total event-driven mode and with a highly efficient thread pooling management.

The Movicon Tags have numerous properties, to render Tags real operating centers. They can be associated with commands to be executed when preset threshold values are reached. They can be set with scaling, realtime connections to database, to OPC servers or clients and with networking modes. Each Tag can be traced with high precision accuracy in a DB with custom messages. The Tag can have fixed device addresses or be kept independent from the drivers as previously. "Structure" type variables can be managed in heterogenous data.

The project tags can be imported directly from the PLC (Simatic S7, Rockwell, Mitsubishi and other..), allowing noticeable time saving. Moreover the project can be associated to dynamic Tag objects, which means those not expressly declared in the current project (distributed Tag database structure).



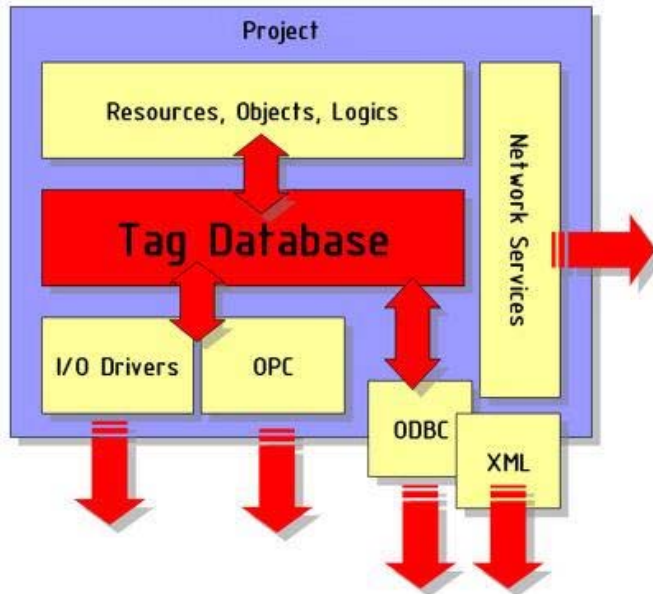
The Movicon variables form the real-time database for all the project resources and executable logic. They can be associated to objects or the project's resources (Screens, Menus, Alarms, Data Loggers, Recipes, Trends, etc.) or combined together by using the logic (Compiled Logic or Basic Script).

Movicon can handle variables in memory areas up to 4 Gigabytes (over 4 billion bytes for each type of variable), according to the maximum quantity of dynamic memory manageable by the operating system. This number is defined as 'Virtually unlimited' when considering that it would be difficult for a multi-complex plant to exceed ten thousand variables.

The variables can be set in '**Shared**' or '**NON Shared**' Movicon areas. When using the '**Shared**' areas it is necessary to define the number of variable bytes to be used in the project, by means of the "**Real Time DB General Settings**".



The number of variables to be exchanged with the field by Movicon (through communication Drivers, OPC, etc.) depends on how many Tags have been enabled by the system's protection key, based on the version purchased.



The project's variables can be allocated through the Movicon Real Time DB and they can also be connected to communication Drivers, to linking functionalities in network (TCP/IP), ODBC links or OPC (OLE for Process Control) links.

Movicon can organize its own variables within a database where they can be assigned an unique mnemonic code (symbolic) and descriptive comment to identify them through out the entire project.

The Variables Real Time DB also permit:

1. The project's local variables to be dynamically linked to the plant's variables through communication Drivers
2. The project's local variables to be dynamically linked to remote variables of Servers in net, using the TCP/IP protocol networking potentialities
3. The project's local variables to be dynamically linked to data fields in Database files using the ODBC functionalities
4. The variables to be dynamically linked to OPC Client applications by means of the OPC standards. The Variables DB can carry out OPC Server functions, while the project's OPC Editor permits links as OPC Client



The Movicon proprietary technology ensures that the increase in the number of variables inserted into the Real Time DB will not effect system performances due to the fact that Links to variables are executed during the startup of the project in Runtime only. This enables Movicon to manage huge amounts of variables in the Real Time DB without degrading system performances.

For further information about **"Communication Drivers"** please refer to the specific section.

# Variable List (Tags)

The 'Variable List (Tags)' resource is used for declaring which variables are to be used within projects for the logical, for the Resources, Controls, Communication drivers etc. Each variable must have a unique name within the 'Variable List (Tags)' resource, which will be recognized within the project by its symbolic name. When using Movicon "Shared" data areas to generate variables, these will also have an absolute address, but the variables' symbolic names will always refer to the project's internal. The absolute address can be changed after without jeopardizing the variables correct identification in points of the project where they are being used.

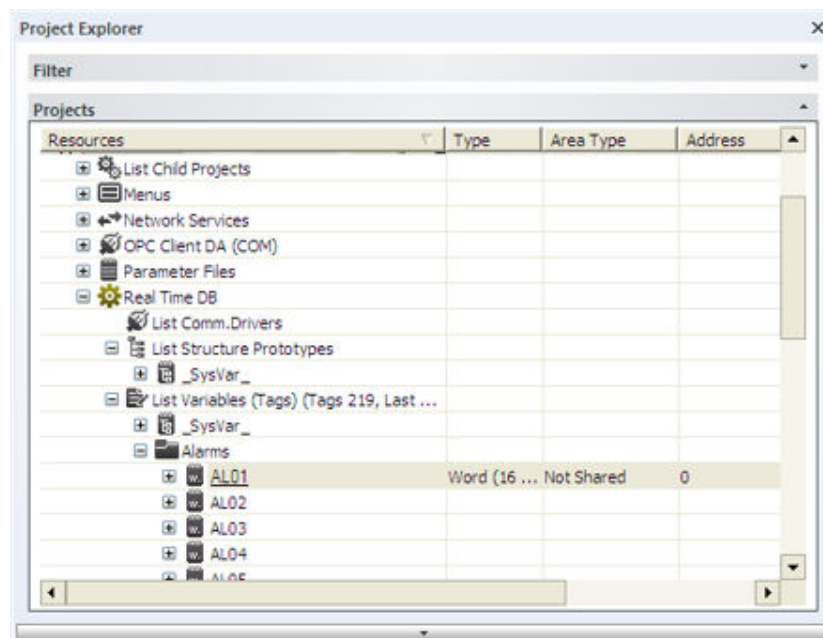
The Movicon 'Variables List (Tags)' are 'smart' types, which means Alarms, Data Loggers or Recipes can be associated to single variables. In addition to this each variable can be enabled with the Tracer, OPC, Networking, Scaling functions etc., through the 'Property Windows'.



Using the 'Variables list' will not effect system's performance, as links to variables take place all at once during the startup of the project's runtime. This technology does not sacrifice feedback times even when extended databases with thousands of variables are being managed.

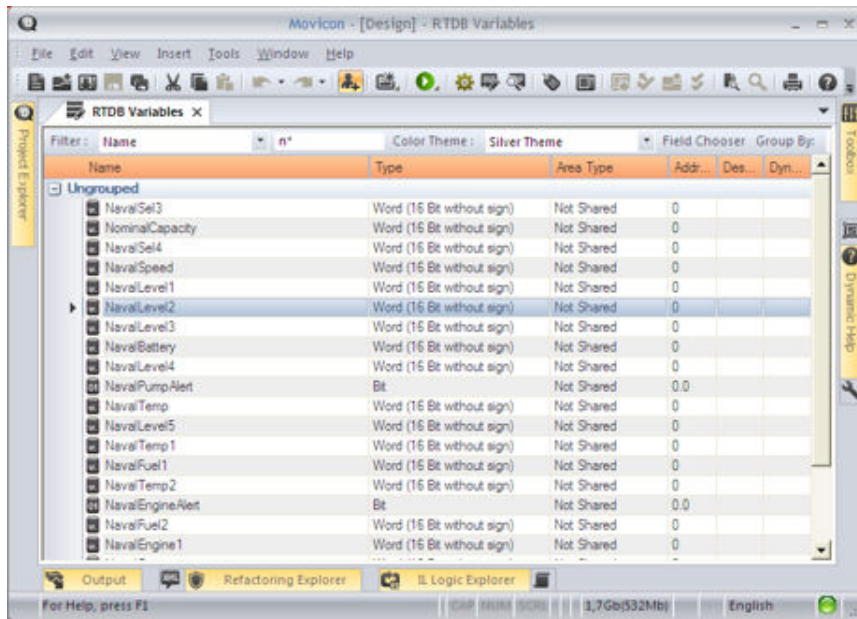
The advantages offered by using the 'Variables List (Tags)' are:

1. reduction in typing or addressing errors in logic programming
2. reduction in program editing time of varied logic addresses when using Movicon "Shared" data areas. Not necessary to modify project when changing absolute addresses of variables
3. direct association of events to variables of the Realtime DB such as Alarms, Data Loggers and Recipes
4. direct reading/writing of remote data on network Servers with Networking functionalities using TCP/IP standard protocol
5. direct reading/writing of data on Database files using ODBC
6. OPC Client application information availability due to the OPC Server function integrated in Variables' properties
7. variable Tracer option settings
8. Scaling option settings of the data associated to the variable



To get a clearer and more readable display of the Variable list, defined in the project's Real Time DB, you can use the "List Variables" window which can be easily opened by double clicking on the "List Variable(Tags)" resource.





You can create Local Variables inside a Screen resource which are created and destroyed together with the Screen and which can be used only within the context of the Screen.

## Variable Tracing (Audit Trail)

Variable Tracing is a Movicon functionality which allows you to historically log the behaviour of variables by recording on database files or text file every change taken place and the cause which changed the variables' values. This is done to keep track of the values which the variable changes to during a process run and the causes which modify the variables' values.

By enabling this functionality Movicon will go and monitor, internally, the required variable's value and will record in the Database or text file, dedicated for this purpose, each single variation of the value contained in the variable when it happens. The file containing information on the Variable tracing will be created automatically by Movicon in the "DATA" folder of the projected called "NameProject\_TraceDB.mdb" in Access2000 format Access2000 or "VariableName.dat" in text format. However you can customize the file name and ODBC link through the "**Real Time DB Trace DB Settings**".

of the 'Real Time DB' resource. After which a table for every variable enabled with the Tracer will be created in the Database. Therefore for each variable enabled with the Tracer Option a Table will be generated inside the Database to contain a Record for each variation endured by the variable, with a number of columns which report the following information:

- **TimeCol:** indicates the recording's date and time in GMT
- **LocalCol:** indicates the recording's data and time in local time
- **MSecCol:** indicates the recording's time in milliseconds
- **UserCol:** indicates the active user's name when recording took place
- **ActionCol:** indicates which event changed the variable, eg. a Screen object, communication driver, watch window etc. If the variable being Traced is a structure type variable, the name of the Member Variable which underwent the change, will also be reported in this field
- **BeforeCol:** indicates the value just before the variable was modified
- **AfterCol:** indicates the variable's new value
- **ValueCol:** indicates the variable's current value
- **QualityCol:** indicates the variable's quality status

To enable this functionality, you need to set the "Variable Trace Options Proprieties" through the variable's Property Window. To enable the Tracer function in more than one variable all at once, just execute a multiple selection of the required variables in the 'Variable List (Tags)' list and enable the "Variable Trace Options Proprieties" through the 'Property Window' which will be activated in all the selected variables.

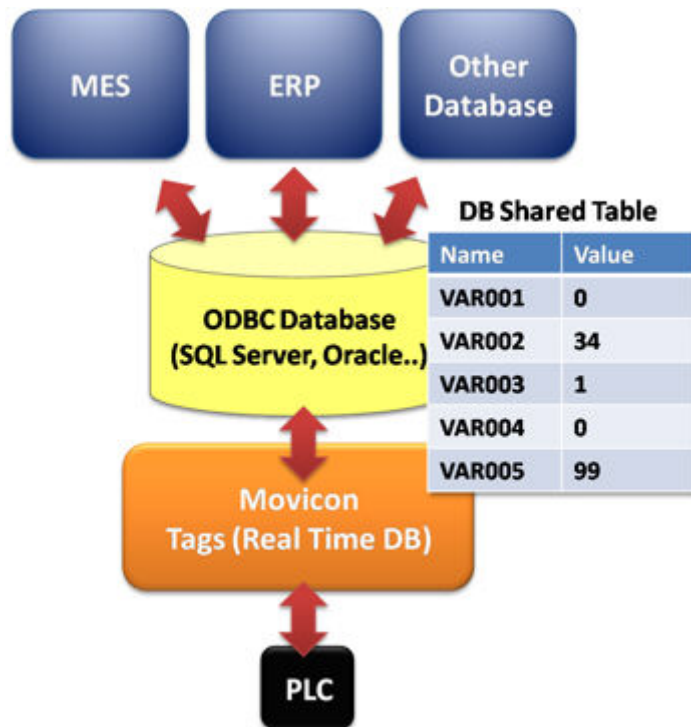
You can display Trace data in table formats through the appropriated TraceDB Window which can be inserted into any project screen. The operator can use this window to verify values which have been recorded for each variable enabled with the tracer.



A TraceDB Window can be dedicated to displaying remote data from a Server connected in network through the Networking facilities.

## Variable Sharing with ODBC

This feature consents database variable value and statistical data sharing. This is aimed at, for instance, to render variable data available to other applications capable of reading/writing Database files. To enable this function you will need to configure the variable's "ODBC Real Time I/O Link" properties and customize the name of the table and ODBC link by means of using the "ODBC Real Time I/O Link Settings" if need be.



Movicon, if not specified otherwise, will create a table in the selected database format in the name of "RTVar". A record for each variable enabled for database sharing will be reserved within this table. The significance of each table column are as follows:

- **Name:** "nvarchar" data type field. This field shows the name of the variable
- **Val:** "nvarchar" data type field. This field shows the current variable value
- **MinVal:** "float" data type field. This field shows the variable's minimum obtained value
- **MaxVal:** "float" data type field. This field shows the variable's maximum obtained value

- **AveVal:** "float" data type field. This field shows the variable's average obtained value
- **TotTime:** "float" data type field. This field shows the total ON time in seconds (time in which the variable last remained at a value that was not zero)
- **LastTime:** "datetime" data type field. This field shows the variable's last ON Time (the time in which the variable last obtained a value that was not zero)
- **MinValDay:** "float" data type field. This field shows the minimum value reached by the variable during the current day
- **MaxValDay:** "float" data type field. This field shows the maximum value reached by the variable during the current day
- **AveValDay:** "float" data type field. This field shows the average value of the variable during the current day
- **TotTimeDay:** "float" data type field. This field shows the total time ON of the variable (the length, in seconds, of the time period the variable had a non zero value during the current day)
- **LastTimeDay:** "datetime" data type field. This field shows last time the variable was ON during the current day (the time when the variable had a non zero value for the last time)
- **MinValWeek:** "float" data type field. This field shows the minimum value reached by the variable during the current week
- **MaxValWeek:** "float" data type field. This field shows the maximum value reached by the variable during the current week
- **AveValWeek:** "float" data type field. This field shows the average value of the variable during the current week
- **TotTimeWeek:** "float" data type field. This field shows the total time ON of the variable (the length, in seconds, of the time period the variable had a non zero value during the current week)
- **LastTimeWeek:** "datetime" data type field. This field shows last time the variable was ON during the current week (the time when the variable had a non zero value for the last time)
- **MinValMonth:** "float" data type field. This field shows the minimum value reached by the variable during the current month
- **MaxValMonth:** "float" data type field. This field shows the maximum value reached by the variable during the current month
- **AveValMonth:** "float" data type field. This field shows the average value of the variable during the current month
- **TotTimeMonth:** "float" data type field. This field shows the total time ON of the variable (the length, in seconds, of the time period the variable had a non zero value during the current month)
- **LastTimeMonth:** "datetime" data type field. This field shows last time the variable was ON during the current month (the time when the variable had a non zero value for the last time)
- **MinValYear:** "float" data type field. This field shows the minimum value reached by the variable during the current year
- **MaxValYear:** "float" data type field. This field shows the maximum value reached by the variable during the current year
- **AveValYear:** "float" data type field. This field shows the average value of the variable during the current year
- **TotTimeYear:** "float" data type field. This field shows the total time ON of the variable (the length, in seconds, of the time period the variable had a non zero value during the current year)
- **LastTimeYear:** "datetime" data type field. This field shows last time the variable was ON during the current year (the time when the variable had a non zero value for the last time)

This information consents reports to be made for preventive maintenance for instance. The "TotTime" column (time in which the variable remained at value that was not zero) is Float type and shows the value in seconds. This field is updated only when the variable returns to the zero value. The "LastTime" column (the time in which the variable last obtained a value that was not zero) shows the date and time the variable obtained a value that was not zero. This field is zeroed by obtaining the "1900-01-01 00:00:00.000" value when the variable returns to the value zero. The variable's TotalTimeOn value is calculated based on the "LastTime" and "TotTime" columns. If the "LastTime" column has a value

higher than "1900-01-01 00:00:00.000", then the variable's TotalTimeOn will be equal to the value of the "TotTime" column plus the number of seconds between the current time and the one reported in the "LastTime" field.

**The statistical data shown in the table is taken from the Retentive Variables file. To get all the above described data correctly, the shared database variables must be enabled in both the "Retentive not Shared" and "Enable Statistic Data" properties.**

## Structure Variables

---

Structure variables are managed by the Real Time DBMS as byte arrays, therefore in the shared database a String field is created where the values of each array item will be shown.



**The string members of structure variables are not shared in the database. The strings in structures are managed as internal variables and have changeable sizes. As structures in the Real Time DBMS are managed as data byte arrays, it is not possible to read/write structure variable string members. Nevertheless, each string will be allocated with two bytes, even though they may not get used.**

**For instance, if a structure variable, composed of two Word members and a String member, is shared in the ODBC, the byte array to be shared in the database will be composed of 6 bytes (4 bytes for the Word members and 2 bytes for the String members).**

## Copy&Pasting Variables

---

When the copy&paste is used for Movicon variables, a smart numbering procedure is used upon pasting. When a variable is added it is actually copied increased, or ending with the next index if necessary. For instance, when you copy the "VAR00001" variable, it will be increased to "VAR00002" when pasted and any other copied variables will follow suit when pastes, in this case the next copied variable will be pasted as "VAR00003". However, if a "VAR00002" variable already exists, then the copied "VAR00001" variable will be pasted with the next available index.

Furthermore, you can also copy&paste variable groups and subgroups. The variable group tree is not limited to just one level but subgroups can be created with unlimited nodes. In any case, variables must have their own unique name within the RealTime DB, independently from the group they belongs to, meaning that no two variables can have the same name even when belonging to different groups.

## System Variables

---

Movicon has a range of prefixed variables, dedicated to give the programmer the possibility to interact with the project and with the system, to facilitate information availability and managed commands by exploiting the logic.



**The System Variables managed in Words contain a number in decimal format supplied or to be supplied to the system.**

**The "strobe" bits are managed by using the logic state set at '1' for safe task synchronizing. When the strobe is 'from Movicon to Logic', the supervisor will set the strobe bit to '1' and the logic, after having interpreted it, will return it to state '0'. When the strobe is 'from logic to Movicon', the logic should set the strobe bit to '1' and the supervisor, after having interpreted the request, will reset the state to '0'.**

The System variables are grouped into a variable structure type. To use the System Variables you first need to insert the variable structure by using the **'Add a System Variable'** command in the **"Project Explorer"** window or by right mouse clicking on the **'Real Time DB'** resource. By executing this command the structure prototype and the relative variable called **"\_SysVar\_"** will be inserted. Each member of the structure variable has a precise significance as described in the table below:

Variable Name	Type	Description
SimSinDouble	Double	Simulation of sine of an angle varying from 0 to 360 degrees. The resulting value will range from -1 to +1 including decimals.
SimSinInt	Sign Byte	Simulation of sine of an angle varying from 0 to 360 degrees, The value is express in percentages as an integer and will range from 100 to +100.
SimCosDouble	Double	Simulation of cosine of an angle varying from 0 to 360 degrees. The resulting value will range from -1 to +1 including decimals.
SimCosInt	Sign Byte	Simulation of cosine of an angle varying from 0 to 360 degrees. The value is expressed in percentages as an integer and will range from 100 to +100
SimRampDouble	Double	Simulation of a saw-tooth ramp with values ranging from -10 to +10 including decimals.
SimRampInt	Sign Word	Simulation of a saw-tooth ramp with values ranging from -10000 to +10000.
SimRandDouble	Double	Random number generation with values ranging from 0 to +32767.
SimRandInt	Sign Word	Random number generation with values ranging from 0 to +32767.
SimRandString	String	Random string Generation.
StrobeLocalTime	Bit	Command Strobe for carrying out changes to the system's time. Movicon executes the update and automatically returns the command to zero value.
ToggleBits	Byte	Each bit of this variable blinks with a different frequency: Bit 00 = 125 ms Blinks Bit 01 = 250 ms Blinks Bit 02 = 500 ms Blinks Bit 03 = 1 s Blinks Bit 04 = 2.5 s Blinks Bit 05 = 5 s Blinks Bit 06 = 10 s Blinks Bit 07 = not used
ActTimeSec	Byte	System Time: Seconds
ActTimeMin	Byte	System Time: Minutes
ActTimeHour	Byte	System Time: Hours

ActTimeDay	Byte	System Time: Day
ActTimeMonth	Byte	System Time: Month
ActTimeYear	Word	System Time: Year
ActTimeString	String	System Time in string value
ActDateString	String	System Date in string value
StrobeExitApp	Bit	Exit from Movicon
StrobeYearLocalTime	Word	Year to be set in System's time (> 1980)
StrobeMonthLocalTime	Byte	Month to be set in system's time (1-12)
StrobeDayLocalTime	Byte	Day to be set in system's time (1-31)
StrobeHourLocalTime	Byte	Hour to be set in system's time (0-23)
StrobeMinuteLocalTime	Byte	Minutes to be set in system's time (0-59)
StrobeSecondLocalTime	Byte	Seconds to be set in system's time (0-59)
CommDriverStatus	Bit	Driver Communication Status
ControlKeyDown	Bit	CTRL key pressed
AltKeyDown	Bit	ALT key pressed
ShiftKeyDown	Bit	SHIFT key pressed
LastKeyPressed	Word	Code of the last key pressed
ActiveScreen	Word	The ID of the currently active screen is set in this variable. In order to use this variable correctly you need to associate each Screen with that different ID.
NumberNotAckAlarms	Word	Number of alarms still active but not yet acknowledged.
AlarmsActive	Word	Number of active alarms. Alarms which are still active, in the ON condition, are considered whether already acknowledge or still to be acknowledged.
AlarmsSoundState	Bit	Variable which sets the sound status of the alarms. This variable can also be set for activating/deactivating the sound by logic.
MouseMove	Bit	This variable when set at true indicates that the Mouse is moving. This is not managed In Windows CE as touch panel devices do not have the mouse.
OutputAckAlarms	Bit	This variable is set at True by Movicon when alarms are acknowledged with the button in the alarm window. The programmer must

		reset the variable to false.
OutputResetAlarms	Bit	This variable is set at true by Movicon when alarms are reset with the button in the alarm window. The programmer must reset the variable to false.
InputAckAlarms	Bit	This variable acknowledges alarms when set at true. After the command has been executed Movicon will reset the variable to false.
InputResetAlarms	Bit	This variable resets alarms when set at true. After the command has been executed Movicon will reset the variable to false.
ActiveUserName	String	This variable contains the name of the active user. Its value will be nothing when no users are logged on.
ActivePassLevel	Sign Word	This variable contains the active user's password level. Its value will be -1 when no user is logged on.
ActiveMaskLevel	Signed DWord	This variable contains the active user's access level. Its value will be -1 when no user is logged on.
LastAlarmText	String	In this variable are shown all the active alarm of the project with a scroll of three seconds. The scroll will be executed following the activation time of the alarms, from the oldest one to the newest one. The scroll will be executed only for the alarms with a status "ON" and not yet acknowledged.
DispatchingQueueCount	Word	All the alarms are displayed with a scroll lasting around 3 seconds. Scrolling is done according to the activation alarm sequence, being from the oldest to the newest. Only alarms with the ON status that have NOT yet been acknowledged are scrolled.
StrobeLoadScreen	Bit	The screen indicated by the _SysVar_:ScreenToLoad variable is loaded on strobe event when not at zero. Once the screen is loaded, Movicon will zero the _SysVar_:StrobeLoadScreen automatically.
ScreenToLoad	Word	Indices which is the ID of the screen to be opened when the StrobeLoadScreen command is given. Each screen can in fact be identified by a ID in their General properties. You can also open child project screens providing that they have a different ID from all the other Child and Parent projects, otherwise all those with the same ID will open. You can use the Parent project's system variable with or without the <..\> prefix in the child project.
HostType	String	Contains the PC's network name. Updates at project StartUp.

HostIP	String	Contains the computer's IP address. Updates every minute.
PlatformVerMajor	Word	Identifies the Movicon Major version (ie. PlatformVerMajor = 11 in the Movicon 11.0.1008). Updates at project StartUp.
PlatformVerMinor	Word	Identifies the Movicon minor version (ie. PlatformVerMinor = 0 in the Movicon 11.0.1008 ). Updates at project StartUp.
PlatformVerBuild	Word	Identifies the Movicon Build number (ie. PlatformVerBuild = 1008 in the 11.0.1008 version). Updates at project StartUp.
PlatformType	String	Contains the name of the platform on which the project runs. This may be "MOVICON", "MOVCE", "POWERHMI", etc..
StartupTime	String	Contains the date and time of the project startup. Updates at project StartUp.
CPUNameString	String	Contains the name of the CPU (ie. Intel(R) Pentium(R) 4 CPU 3.00GHz). Updates at project StartUp.
CPUVendorIdentifier	String	Contains the name of the CPU vendors (ie. GenuineIntel). Updates at project StartUp.
CPUIdentifier	String	Contains the CPU version, family, model, step (ie. x86 Family 15 Model 3 Stepping 3). Updates at project StartUp.
CPU Speed	String	Indicates the CPU's speed (ie. 2993). Updates at project StartUp.
WinMajor	String	Indicates Windows' Major version number (ie. 5). Updates at project StartUp.
WinMinor	String	Indicates Windows' Minor version number (ie. 1). Updates at project StartUp.
BuildNumber	String	Indicates the Windows build (ie. 2600). Updates at project StartUp.
ServicePack	String	Indicates the Windows Service Pack version number (is. 2). Updates at project StartUp.
OSType	String	Indicates which Operating System edition type (ie. Windows XP Professional Edition Service Pack 2). Updates at project StartUp.
TimeZoneBias	String	Indicates the local time zone in minutes considering the daylight saving time.
IntColumnSep	String	Indicates the decimal separator character for Numbers set in the OS's International Options.
IntTimeSep	String	Indicates the separator character for the time set in the OS's International Options.
IntDateSep	String	Indicates the separator character of the data set in the OS's International Options.



ShortDateFmt	String	Indicates the short date format to display the date set in the OS's International Options.
TimeFmt	String	Indicates the time format to be display the date set in the OS's International Options date settings.
WorkingDir	String	Contains the directory's current set value.
LoginUserName	String	Contains the name of the user currently logged in the OS.
MemoryLoad	Word	Indicates the percentage of allocated RAM.
MemoryTotalPhys	Double	Indicates the total of physical RAM in the system (in Bytes).
MemoryAvailPhys	Double	Indicates the quantity of physical RAM available in the process (in Bytes).
MemoryTotalPageFile	Double	Indicates the maximum memory limit that can be allocated in the system (in Bytes).
MemoryAvailPageFile	Double	Indicates how much available memory not yet allocated in the system (in Bytes). The difference between MemoryTotalPageFile and MemoryAvailPageFile is the memory allocated in the system.
MemoryTotalVirtual	Double	Indicates the total virtual memory available in the system (in Bytes).
MemoryAvailVirtual	Double	Indicates the amount of virtual memory not yet used in the system (in Bytes).
FreeSpaceAlarmPath	Double	Indicates available space in KBytes on the disk containing the project's Alarms folder set in the "Project Paths" settings properties.
FreeSpaceResourcePath	Double	Indicates the space available in bytes on the disk containing the project's resource folder set in the "Project Paths" settings properties.
FreeSpaceImagePath	Double	Indicates the space available in KBytes on the disk containing the project's Images file set in the "Project Paths" settings properties.
FreeSpaceLogPath	Double	Indicates the space available in KBytes on the disk containing the projects Logs folder set in the "Project Paths" settings properties.
FreeSpaceDLRPath	Double	Indicates the space available in KBytes on the disk containing the project's Data Logger/Recipe folder set in the "Project Paths" settings properties.
FreeSpaceNetworkUserPath	Double	Indicates the space available in KBytes on the disk containing the project's Networking log folder set in the "Project Paths" settings properties.

FreeSpaceDataUserPath	Double	Indicates the space available in KBytes on the disk containing the project's Data folder set in the "Project Paths" settings properties.
ILLogicCycleTime	Word	The project's IL logic cycle time.
PeekInUseByte	Word	retrieved peak of bytes in use during project run.
ProjectRunning	Bit	Indicates whether project is running. The "True" value means that the project is being Run.
TotalInUseByte	Word	Total number of Bytes currently in use in the project.
NumInUseVar	Word	Indicates the number of variables which have gone into use but are still pending because Movicon has not yet processed them. This value changes from zero to a different value when the variables go into the "In Use" from the "Not in Use" status, usually at project startup or page change. Once all the variables have been processed this value returns to the zero value. When the project does not have a high number of variables, this value will stay with a different value from zero for a very short time and therefore not always retrievable.
NumDynVar	Word	Number of variables created dynamically for objects set with a dynamic link in their "Variable" property.
SRActiveShortcut	String	Speech Recognition: contains the path and name of the Shortcut resource currently active in the project. (ie. C:\ProjectName\Resources\ProjectName\Shortcut1.movacc). If the active Shortcut has not been enabled with the Speech Recognition functions, the variable will be an empty string.
SRLastUnderstood	String	Speech Recognition: contains the last word or phrase understood by the system.
SRListening	Bit	Speech Recognition: when the value is "True", this means that the Speech Recognition functions have been enabled.
SRIActive	Bit	Speech Recognition: when the value is "True", means that the system is active and listening.

# Variable Areas

---

In Movicon the variables can be mapped in two different data areas, the data area defined 'Shared' and the other defined 'NOT Shared'. The 'Shared' data area, to be managed as it was in the previous Movicon version, is subdivided into three variable areas: Input data Area, Output data Area and Internal data Area (Flag). The 'Shared' data area provides each variable with an absolute memory address associated by the programmer. The 'NOT Shared' data area does not provide any such address for variables which is done by Movicon by allocating the variable to an area unaccessible to absolute addresses.

All the Movicon variables can be managed in the program in bit, byte (8 bit), a word (16 bit), doubleword (32 bit), float (32 bit in floating comma), long (64 bit), string or array. Particular variables, called **"Structure Variables"**, are added to these which are real data structures.



**All the variables, Input Area, Output Area, Flag Area and Non Shared Area can be exchanged with the field.** The only reason that distinction is still made between the three areas for Shared variables is so they can be compatible with old and previous versions.



To avoid overlapping errors of unwanted variables it is advised to always use the 'Not Shared' areas. By doing this you can access to the variable bit using the "VariableName.NumberBit" (for instance typing a text for the '0' bit of the "VAR0001" just type 'VAR0001.0').

When using the "Shared" area, you can create more mapped variables in the same area. This is very handy for setting Word variables used for communicating with the field and 16 bit variables to be used as single items in objects in screens. In this case the Word variable and the Bit variable should be mapped in the same area (ie. Address Word = 0, AddressBit1 = 0.0, AddressBit2 = 0.1, etc). When using this technique you will need to keep in mind that each one of the variables in the mapped shared area (partly or completely) will be put into use in the same area of another variable already in use (therefore, if a Bit variable is put into use, the Word variable will be put into use as well) This behaviour is valid for standard and array variables but not for structure variable members.

## Retentive Variables and Statistical Data

---

You can use retentive variables when the project must store the variables' value after the project has been closed or the hardware platform turned off. The retentive variables can be specified for either the 'Shared' areas (Flag, Output, Input) or for those 'NOT shared'. To use the retentive variables you have to set their modes and quantities through the **"Real Time DB Retentive Data Settings"** for variables of 'shared' areas, and through the **"Retentive not Shared"** property (paragraph: "Variable General Properties") for variables of 'NON Shared' areas.



Any variables of the 'Shared' area declared retentive will start from the starting address to the ending address specified. It is also necessary to always check the congruency between the retentive variable area and the size of the established area. For instance, let's say 100 Outputs bytes are going to be used this means using the area from 0 to 99! The system will give warnings of any incongruities.

The retentive variable files not only keep current variable values saved they can also contain a series of statistical data about the variables. This statistical data only gets saved when the variable's "Enable Statistic Data" property is enabled.

### **"NOT Shared" Retentivity Area**

Movicon will create a file in XML format for each variable defined as retentive belonging to the "NOT shared" area within which the value of the current variable will be saved. These files, which are found in the project's 'DATA' folder, will be named in the following way:

**NameProject\_NameVariable.var**

for example the "VAR0001" variable of the "Project1" project will be in a retentivity file called:

**Project1\_VAR0001.var**

#### **"Shared" Retentivity Area**

However, variables belonging to the "Shared" area need their retentivity settings defined through the "Real Time DB Retentive Data Settings". In this case one retentivity file will be created for each area, Input, Output and Flag, within the "DATA" folder. These files will have the following names:

**ProjectName.FLG:** retentivity file for variables in the "Flag" area  
**ProjectName.FBK:** Backup retentivity file for variables in the "Flag" area  
**ProjectName.IN:** retentivity file for variables in the "Input" area  
**ProjectName.IBK:** Backup retentivity file for variables in the "Input" area  
**ProjectName.OUT:** retentivity file for variables in the "Output" area  
**ProjectName.OBK:** Backup retentivity file for variables in the "Output" area

### **Retentivity File and Statistical Data**

---

The last value obtained by the variable and other statistical data, if enabled, are saved in the retentivity file of each variable. The information available is:

#### **Current Data**

- Actual variable value
- Date of last variable update
- Date of last transition from the zero value to a zero other than zero. This date (LastTimeOn) is only significant when the variable has a value that is not zero, and will be reset when the variable turns back to the zero value

#### **Statistical Data**

- Minimum variable value
- Maximum variable value
- Average variable value
- Total time in which the variable remained at a value different from zero (this data is updated on file only when the variable returned to the zero value)
- Date of the last time the variable obtained a value different from zero (the date is zeroed when the variable returns to the zero value)

The five Statistical Data items described above are calculated starting from the first project run and are zeroed only when the variable's "Reset Statistic" command is used.

#### **Daily Statistical Data**

- Daily minimum variable value
- Daily maximum variable value
- Daily average variable value
- Daily total time in which the variable remained at a value different from zero (this data is updated on file only when the variable returned to the zero value)
- Date of the last time the variable obtained a value different from zero for current day (the date is zeroed when the variable returns to the zero value)

The five Statistical Data items described above are calculated on a daily basis. This means that they will be automatically zeroed at midnight (passing over from the current day at 23:59:59 to 00:00:00 of the next day). The same zeroing will take place if the current day of the project startup is different from when the project was last stopped.

#### **Weekly Statistical Data**

- Weekly minimum variable value
- Weekly maximum variable value
- Weekly average variable value

- Weekly total time in which the variable remained at a value different from zero (this data is updated on file only when the variable returned to the zero value)
- Date of the last time the variable obtained a value different from zero for current week (the date is zeroed when the variable returns to the zero value)

The five Statistical Data items described above are calculated on a weekly basis. This means that they will be automatically zeroed at midnight before the new week begins (passing over from Sunday at 23:59:59 to 00:00:00 of Monday). The same zeroing will take place if the current week of the project startup is different from when the project was last stopped.

#### Monthly Statistical Data

- Monthly minimum variable value
- Monthly maximum variable value
- Monthly average variable value
- Monthly total time in which the variable remained at a value different from zero (this data is updated on file only when the variable returned to the zero value)
- Date of the last time the variable obtained a value different from zero for current month (the date is zeroed when the variable returns to the zero value)

The five Statistical Data items described above are calculated on a monthly basis. This means that they will be automatically zeroed at midnight before the new month begins (passing over from the last day of the month at 23:59:59 to 00:00:00 of the first day of the next month). The same zeroing will take place if the current month of the project startup is different from when the project was last stopped.

#### Yearly Statistical Data

- Yearly minimum variable value
- Yearly maximum variable value
- Yearly average variable value
- Yearly total time in which the variable remained at a value different from zero (this data is updated on file only when the variable returned to the zero value)
- Date of the last time the variable obtained a value different from zero for current year (the date is zeroed when the variable returns to the zero value)

The five Statistical Data items described above are calculated on a yearly basis. This means that they will be automatically zeroed at midnight before the new year begins (passing over from the last day of the year at 23:59:59 to 00:00:00 of the first day of the next year). The same zeroing will take place if the current year of the project startup is different from when the project was last stopped.

## Renaming Variables

---

Movicon has a function which consents to updating all the references to variables in projects automatically when they are renamed. This means that when a variable is renamed you will not need to go and replace the old name with the new one in the various points of the project where the variable has been used.



**The automatic replacement mechanism is active only if the project execution property "Enable Renaming Manager" is enabled. If not, all the aliasing mechanism as described below will not work and the renamed variables should be replaced manually where used in the project.**

Movicon uses an 'aliasing' mechanism which keeps track of the variable's original name and its new one. When the variable is renamed many times, only the first and last name is tracked down. This information, consisting of the relationship between the variable's original name and the last one associated, is contained in the ".movrealtimedb" file with the <RenamedVariables> tag, as shown below:

```

...
<RenamedVariables>
<item key="VAR00001" value="VAR00002"/>
<item key="VAR00003" value="VAR00006"/>
<item key="VAR00004" value="VAR00007"/>
</RenamedVariables>
...

```

where:

**item key:** represents the variable's original name

**value:** represents the last name used for renaming the variable

If the Renaming Manager has been enabled, it is recommended not to use the original names of the renamed variables to create new variables. When trying doing so, Movicon shows an advice message as the following:

Warning! The variable Name 'VAR00001' has been renamed to 'VAR00002' in this project. Do you want to use this variable name? Answering "Yes" will remove the recorded renamed link.

Warning! The 'Screen1' resource name have been renamed in 'Screen2' in this project. Do you wish to use the same resource name? If you ask with 'Yes? the saved rename link will be removed.

The message above has been generated since you tried to create or rename a variable using the "VAR00001" as new name, which has already been used as original name for the variable renamed "VAR00002". Answering "Yes" to the message box, the link "original name - new name" will be deleted and the automatic replacing mechanism will be cancelled for that variable.

As an example, if "VAR00001" has been associated to a display object, then "VAR00001" is renamed as "VAR00002", the display object will result associated to "VAR00002". The RealTimeDB will contain only one variable, that is "VAR00002". If later a new variable is added to the database using "VAR00001" as new name, the renaming link will be deleted and the display object will result associated to "VAR00001", being VAR00001 a new variable. Both VAR00001 and VAR00002 will be present in the RealTimeDB, being two different variables.

When insert a new variable Movicon will propose a name which has not yet been used in the RealTimeDB and which has not yet been used as a original name of a variable which has already been renamed. If, for instance, you insert the VAR00001 and then rename it to VAR00002, when inserting a new variable Movicon will propose VAR00003 as its name. When a variable is renamed, any retentivity file belonging to the variable will not get automatically renamed and as a result this variable will have a "0" value at the project startup and the first modification to this variable will create a new retentivity file with the variable's new name.



**WARNING! The aliasing mechanism can be cancelled deleting from the ".movrealtimedb" file the xml tags listing the renamed variables. This way the renamed variables will keep their last name, while the references added for example to objects or commands will be linked to the variables original names. As a result, the reference to original variables could be invalid.**

The automatic variable aliasing works according to the object or resource in question:

- **Standard objects in screens:** new variable names will appear in properties where variables can be associated
- **Command List Variable commands:** new variable names will appear in properties where variables can be associated
- **Event Objects:** new variable names will appear in properties where variables can be associated
- **Scaling Objects:** new variable names will appear in properties where variables can be associated

- **Scheduler Objects:** new variable names will appear in properties where variables can be associated
- **Shortcuts:** new variable names will appear in properties where variables can be associated
- **Menu:** new variable names will appear in properties where variables can be associated
- **Redundancy:** the new variable name will appear in the "Status Variable" property
- **Screens:** the new variable name will appear in the "Screen Layer Variable" property
- **Alarm Objects:** new variable names will appear in properties where variables can be associated
- **Alarms as Templates:** new variable names will appear in the Alarm Window. If an String ID with the original variable name has been inserted for displaying the alarm's text, you will also need to rename the String ID with the variable's new name
- **Parameter File:** the variable's original name will always appear in the file but the new variable will be used in run mode
- **Trend/Data Analysis:** new variable names will appear in the Trend/Data Analysis properties where variables can be associated. This is also valid for the variables associated to the Trend/Data Analysis pens. However the pens' names will remain unchanged even in those cases when pens have the same variable names
- **Basic Script:** the variable's original name will always appear in the basic script code but the code will be executed according to the new variable in run mode
- **Basic Script Properties:** the new variable name will appear in the "Status Variable" property
- **IL Logic:** the original variable name will always appear in the code but it will be executed according to the new variable in run mode
- **DataLoggers/Recipes:** new variable names will appear in the DataLogger/Recipe properties where variables can be associated. However, the column names will remain unchanged, even in those cases where column names are the same as the variables'
- **Communication Drivers:** original variable names will always appear in Driver Station and Task properties where variables can be associated but the new variable will be used in runtime

## Real Time DB Settings

---

This Settings window is used for customizing the variables management used by the projects in addition to that of the Server OPC, Networking and Variable Tracer Settings. The settings of the number of variables managed by the project are completely free hand to enable you to correctly allocate sufficient memory in function with the number of variables used in the project, adapting the system to the requirements needed. The number of variables to be set only concerns the Movicon Shared memory area and should only be used when really necessary.

### Real Time DB General Settings

---

The general properties allow you to define the maximum number of variables from the Movicon Shared area which can be used in the project.  
To modify the Real-Time DB General property, select the object with the mouse and use the Movicon **'Properties Window'**.

#### Inputs / Outputs / Flags

You have to specify the maximum number of bytes you wish to use for each variable type from the Movicon Shared area managed by the system. The edited number can also be freely set or modified afterwards.

The system request to specify the number of bytes of the variables used in the project permits the memory effectively necessary only to be allocated, consenting the hardware platform being used to optimize the project.

#### **Input Image**

When enabling this selection the synchronization between data read by the communication driver and the project's general logic will be activated. At the beginning of every processing cycle of the general logic a process image of all the project inputs is saved. In this way any updates on inputs, carried out by the communication driver, will not effect the current general logic process but the following general logic process cycle, thus consenting the use of the same input variables in more points in the general logic.

#### **Output Image**

When enabling this selection the synchronization between data written by the communication driver and the project's general logic will be activated. The general logic works with the outputs process image which is loaded into the project's outputs only at the end of each general logic processing cycle. By enabling this selection any undesired triggering on variables of external devices can be avoided when the output variables have been set in more than one point in the general logic.

### **Real Time DB Retentive Data Settings**

---

The Retentive Data property allows you define the areas and which Movicon Shared area addresses must be retentive.

To edit the Real Time DB Retentive Data property, select the object with the mouse and use the Movicon '**Properties Window**'.

#### **Inputs**

This setting allows you to enable or disable the retentivity management for the Input area.

#### **Inputs From:**

This setting permits you to insert the starting address for the retentive Input area.

#### **Inputs A:**

This setting permits you to insert the ending address for the retentive Input area.

#### **Flags**

This setting permits you to enable or disable the retentivity for the Flag area.

#### **Flags From:**

This setting permits you to insert the starting address for the retentive Flag area.

#### **Flags A:**

This setting permits you to insert the ending address for the retentive Flag area.

#### **Outputs**

This setting permits you to enable or disable retentivity management of the Output area.

#### **Outputs From:**

This setting permits you to insert the starting address for the retentive Output area.

#### **Outputs A:**

This setting permits you to insert the ending address for the retentive Output area.

### **Real Time DB InUse Variable Manager Settings**

---

The Variables in Use Management property allows you to optimize performances. A sophisticated architecture consents re-acknowledgement of variables being used by the system and to adapt variable updates to only those that need it.



In this way the system's overall performances will be increased to the advantage to the project's user.

To modify the Real-Time DB Variables in Use Management, select the object with the mouse and use the Movicon '**Properties Window**'.

#### **InUse Manager**

This setting allows you to enable or disable the Variables in use management.

This lets communication with devices to be optimized (by means of Drivers, OPC, etc.) because only the jobs containing the variables in use are kept active. When this property is disabled all the communication jobs will always be kept active and therefore effecting system performances.

#### **Use Shared Dyn. Tag**

When this property is enabled the project's dynamic variables will be allocated to the Real Time DB Shared areas, and in particular the Output area. When this property is left disabled, the dynamic variables will be allocated to the Non Shared areas instead.

## **Real Time DB OPC Server Settings**

---

The Movicon OPC Server can be configured through the appropriated property accessible from the Real Time DB Settings.

The Movicon OPC Server, if registered within the list of OPC Servers available, will startup in automatic, with the last opened project, as soon as the OPC Client connects. To enable this functionality you need to disable the **Unregister Automatic Server** selection.

When selecting the **Automatic Shutdown**, Movicon will close as soon as the OPC Client is disconnected.

To modify the Real-Time DB OPC Server property, select the object with the mouse and use the Movicon '**Properties Window**'.

#### **OPC Server DA**

This selection box enables all the OPC Server's functionalities and has selection priority when this tool is to be used.

#### **OPC Server Name**

This edit box is used for setting the name with which the Movicon OPC Server will be registered with in the operating System.

#### **OPC Server Description**

This edit box is used for setting the description with which the Movicon OPC Server will be registered with in the operating System.

#### **OPC Server AE**

This selection box enables the OPC Server AE being the acknowledge function of the project's alarms and events, to all the OPC Clients connected.

#### **OPC Dynamic Tag**

This selection box is used to enable the possibility to dynamically create links to the project's variables. When checking this option, you can access any of the project's variables from the OPC Client by setting the Tag in the following way: *Name Variable*.

Otherwise only the Real Time DB variables will be show where the OPC Server option is explicitly enabled; please refer to the "Variable Options Properties" document for further information on the OPC Settings of project variables.

#### **OPC Server Debug Trace**

This selection box enables the OPC Server DA to trace any Debug events.

#### **AutoShutDown**

This selection box enabled the automatic shutdown of Movicon when the OPC Client disconnects form the project where the OPC Server is enabled.

#### **AutoRegister**

When this this box is enabled with a check mark, the OPC Server will automatically register at startup.

**AutoUnregister**

This selection box enables the automatic elimination of the Movicon OPC Server from list of available Servers when the project closes.

**Register OPC Server**

Command for registering the OPC Server.

**Unregister OPC Server**

Command for deleting the OPC Server registration.

**Refresh Rate (ms)**

This edit box permits you to set the time of the Tags updating by the Movicon OPC Server DA. Values are to be comprised of milliseconds between 10 and 1000.

*The time set here is to be differentiated from the refresh time set in the property of a group inserted within any OPC Client. It is not advised to set fast refresh times when long times have been set inside the groups created in the Client.*

**AutoShutDown Timeout (ms)**

This setting permits you to insert the OPC Server's standby time before shutting down.

**AE Ack**

This selection box enables the acknowledge function of the events generated.

**Free threaded**

Enables OPC Server model for managing Threads.

## Real Time DB OPC Server Security Settings

---

Security levels for any eventual Clients wishing to connect, can be setup by means of using the OPC Server Security Settings.

To edit the Real-Time DB OPC Server Security properties, select the object with the mouse and use the Movicon **'Properties Window'**.

**Enable Security**

This property allows you to enable the OPC Server Security management. When enabled only users acknowledged by the Server, to which the users connect to through the Client, can carry out operations consented by their access rights.

**OPC Server Min.Impers.Level**

This property permits you to set the minimum Impersonalized access level to be assigned to users who connect to the Server and do not need to be acknowledged.

## Real Time DB OPC XML DA Server Settings

---

The Movicon OPC XML DA Server are configured by using the appropriated properties which are accessed through the Real Time DB settings.

The Movicon OPC XML DA Server, when enabled, makes the OPC XML DA Server available on the system's OPC Server list.

To edit the Real Time DB OPC XML DA Server settings, select the object with the mouse and use the Movicon **"Properties Window"**.

**Enable**

This selection box enables all the OPC Server XML DA functions.

**Transport**

The transport type to be used for Server/Client communication is selected in this box.

**Port**

The port socket number to be used.

**Accepted # Threads**

The number of thread to be left on hold for the new connections.

**Thread Pool Min**

Minimum number of Threads in a group.

**Thread Pool Max.**

Maximum number of Threads in a group.

**Request Backlog**

Number of backlog connection requests before the system starts to refuse connections.

**Request Timeout**

Timeout used when the Server responds to the Client. This value is expressed in milliseconds.

**Request Buffer Size**

Buffer Size used for receiving input messages. It would be best to use the default value as smaller values may effect performances and bigger values may occupy too many resources.

**Keep Alive**

When this property is enabled the connection will not close once a response has been sent to the Client.

**Vendor Info**

This allows you to insert a written comment regarding information on the OPC XML DA Server supplier.

**Enable Security**

Allows you to enable Server OPC XML DA access security.

**Default Access Level**

Allows you to enter the Default Access Level for anonymous connections.

## Real Time DB Trace DB Settings

---

The Variable Tracing is a Movicon functionality which allows you to historically log the behaviour of variables, by recording on Database files or text file each variable change and the cause changing the variable's value. This will enable you to keep track of the value changes that take place to variables and what were the causes. To modify the Real-Time DB Trace properties, select the object with the mouse and use the Movicon '**Properties Window**'.

**Use IMDB Manager**

See paragraph "**IMDB Settings for Recording data**" in the section on "**IMDB**".

**Shared Tables**

See paragraph "**IMDB Settings for Recording data**" in the section on "**IMDB**".

**Save XML File**

See paragraph "**IMDB Settings for Recording data**" in the section on "**IMDB**".

**Crypt File**

See paragraph "**IMDB Settings for Recording data**" in the section on "**IMDB**".

**Write Behind Delay**

See paragraph "**IMDB Settings for Recording data**" in the section on "**IMDB**".

**Max Rows Nr.**

See paragraph "**IMDB Settings for Recording data**" in the section on "**IMDB**".

**Keep the DB Connection open**

See paragraph "DBMS Settings for Recording data on DataBase" in the section on "Historical Log Management".

**Send Admin.Alerts**

See paragraph "DBMS Settings for Recording data on DataBase" in the section on "Historical Log Management".

**Max. Error Number**

See paragraph "DBMS Settings for Recording data on DataBase" in the section on "Historical Log Management".

**Max. Transactions**

See paragraph "DBMS Settings for Recording data on DataBase" in the section on "Historical Log Management".

**Max. Cache Size**

See paragraph "DBMS Settings for Recording data on DataBase" in the section on "Historical Log Management".

**Max. VarChar Precision**

See paragraph "DBMS Settings for Recording data on DataBase" in the section on "Historical Log Management".

**ODBC DSN**

See paragraph "DBMS Settings for Recording data on DataBase" in the section on "Historical Log Management".

**ODBC DSN User**

See paragraph "DBMS Settings for Recording data on DataBase" in the section on "Historical Log Management".

**Time Column**

This setting is used for inserting the Trace Table's Time Column's name. If not specified the default's name will be used instead. The Time column indicates the recording's date and time referring to GMT.

**Local Time Column**

This setting is used for inserting the Trace Table's Local Time Column's name. If not specified the default's name will be used instead. The Local Time column indicates the recording's dates and Time referring to local time.

**MSec Column**

This setting is used for inserting the Trace Table's MSec Column's name. If not specified the default's name will be used instead. The MSec Column indicates the milliseconds relating to the recording time.

**User Column**

This setting is used for inserting the Trace Table's User Column's name. If not specified the default's name will be used instead. The User Column indicates the name of the active user when the recording took place.

**Changer Column**

This setting is used for inserting the name of the Trace Table's Changer Column's name. If not specified the default's name will be used instead. The Changer Column indicates by which event caused the variable's status change, eg. by a screen's object, by the communication driver, by the watch window, etc.

**Value Before Column**

This setting is used for inserting the Trace Table's Value Before Column's name. If not specified the default's name will be used instead. The Value Before Column indicates what the variable's value was before it was just last changed.

**Value Next Column**

This setting is used for inserting the Trace Table's Value Next Column's name. If not specified the default's name will be used instead. The Value Next Column indicates the variable's new value.

**Value Column**

This setting is used for inserting the Trace Table's Value Column's name. If not specified the default's name will be used instead. The Value column indicates the actual value that the variable was meant to be modified with. It may happen that the intended value is not always set exactly the same on the variable. The variable may undergo scaling or conversion which will therefore cause the inserted value to adapt. When this is the case the values reported in the 'Value Column' and the 'Value Next Column' tend to be different.

**Quality Column**

This setting is used for inserting the Trace Table's Quality Column's name. If not specified the default's name will be used instead. This column indicated the variable's quality status.

**Recreate All**

The command regenerates all the tables of the variables enabled with the Trace. All existing data will be lost.

## ODBC Real Time I/O Link Settings

---

The **ODBC Real Time I/O Link Settings** consents the setting up of dynamic links between the Movicon Real Time DB and database file data fields in the preferred DB format.

To modify the ODBC Real-Time DB settings, select the Real Time DB group in the 'Project Explorer' window with the mouse and use the Movicon **'Properties Window'**.



The Database file will be organized with a column containing the names of variables (defined in the name box in the variables' General Properties) and a series of columns containing their current values and corresponding statistics.

When using MS Access as the preferred database, Movicon will create the Database file in the project's "DATA" folder for default with the following name:

ProjectName\_RealTimeDB.mdb

To change the ODBC link and the linked file use the **"ODBC DSN"** properties described below:

**Keep the DB Connection open**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

**Send Admin.Alerts**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

**Max. Error Number**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

**Max. Transactions**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

**Max. Cache Size**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

**Max. VarChar Precision**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

**ODBC DSN**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

**ODBC DSN User**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

**Table Name**

This setting is used for inserting the Table's name. If nothing is entered, the **"RTVar"** default name will be used instead.

**Name Column**

This setting is used for inserting a name for the DBMS table's Variable Name column. This column reports the name of the variables enabled with the ODBC Real Time I/O Link function. Each record in the table represents a variable. If nothing is entered the **'Name'** default name will be used instead.

**Value Column**

This setting is used for inserting the name of the DBMS table's Variable Value Column. This column reports the variable's value enabled with the ODBC Real Time I/O Link function. If nothing is entered, the **"Val"** name for default will be used instead.

**Min. Value Column**

This setting permits you to insert the name of the Min. Value Column of variables in the DBMS table. This column shows the minimum value of variables enabled with the ODBC Real Time I/O Link function. The **"MinVal"** default name will be used if left blank.

**Max. Value Column**

This setting permits you to insert the name of the Max. Value Column of variables in the DBMS table. This column shows the maximum value of variables enabled with the ODBC Real Time I/O Link function. The **"MaxVal"** default name will be used if left blank.

**Ave. Value Column**

This setting permits you to insert the name of the Ave. Value Column of variables in the DBMS table. This column shows the average value of variables enabled with the ODBC Real Time I/O Link function. The **"AveVal"** default name will be used if left blank.

**Total Time ON Column**

This setting permits you to insert the name of the Total Time ON Column variable from the DBMS table. This column shows the total ON time of the variables enabled with the ODBC Real Time I/O Link function. The **"TotTime"** default name will be used if left blank.

**Last Time ON Column**

This setting permits you to insert the name of the Last Time ON Column Variable of the DBMS Table. This column shows the last Time ON of variables enabled with the ODBC Real Time I/O Link function. The **"LastTime"** default name will be used if left blank.

**Recreate All**

This command regenerates the tables containing data of the variables enabled with the ODBC Real Time I/O Link functionality. Any existing data will be lost.

## Dynamic Variables

---

The dynamic variables have been designed to make it easier for the programmer in building screens or anything else exploiting the use of remote variables. By using the dynamic variables you can connect to tags configured in OPC Servers, local or network, to variables belonging to other Movicon projects executed in remote by exploiting the Networking services, and to field devices through communication drivers or OPC tags etc. The dynamic variables can be used in all the controls and drawings in the screens. For example: in the drawing animation properties, in Buttons, etc. The use of dynamic variables is also consented internal Script codes by using the appropriate syntaxes

described in the specific paragraphs ("Dynamic Network Variables", "Dynamic OPC Variables", "Dynamic Communication Driver Variables"). In addition to this, you can also use dynamic variables internal commands belonging to menu items and Shortcut keys. The dynamic variables cannot be used internal General Logic or SoftLogic, also because these are also meant for writing or reading remote variables only in certain moments when absolutely necessary: when they are no longer in use Movicon actually frees them from system memory.

When using dynamic variables you are permitted to go over the number of tags enabled by the license which is being used for running the project. When opening a resource, in which dynamic variables are being used, Movicon automatically allocates these variables in memory, to then frees them from memory when the resource is closed. This makes it theoretically possible to have an unlimited number of dynamic variables within a project.

The dynamic variables are subdivided into three categories: Network Variables, OPC Variables and Communication Driver Variables. Their management by Movicon is identical for all categories.



The number of dynamic variables used must always be compatible with the number of tags provided by the Movicon license being used.

## Dynamic Network Variables

---

The Dynamic Network Variables are needed for creating dynamic connections with remote Movicon projects, connected by means of the Networking functions, commonly known as the Server project. Dynamic network variable selecting can easily be done from the **"Network"** tab in the window which appears after double-clicking on the Variable selection box from any Movicon component or resource.

In order to be able to select Dynamic Network variables directly you need to have the remote computer connected in network and the Server project running.

You can also specify the name or TCP-IP address of the Server in the **"Server"** edit box in the window shown above.

All the considerations made for the Networking also apply when using the dynamic network variables: the connection modalities exploit the same technology.

The **"Refresh"** key can have diverse functions according to the element highlighted in the left square of the window:

- When the local computer or a network computer is highlighted, the **"Refresh"** button updates the list of all the Movicon projects running on that specific computer
- When a domain or a network group is highlighted, the **"Refresh"** button updates the list of computers available for the network highlighted
- When a Movicon Project is highlighted: the **"Refresh"** button updates the list of variables displayed in the square on the right, by applying the filter. The filter consists of alphanumeric characters which can be preceded or followed by a "\*" wildcard character. For instance, filters may be: VAR\*, \*000\*, \*05

If you do not have a remote computer connected in net you can write the correct syntax directly into the Variable insertion box, as follows:

**[NET]\\ComputerName\\VariableName**

where:

**[NET]** = Movicon suffix which identifies a network connection

**ComputerName** = Name of remote Server computer

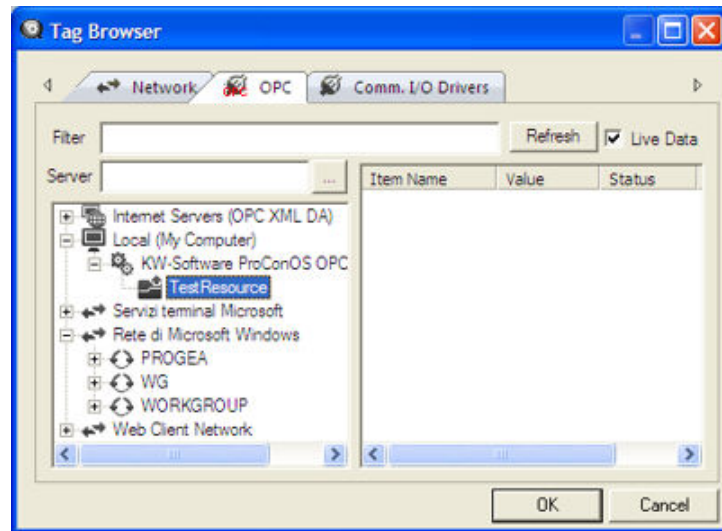
**VariableName** = Name of variable in the Server project to be connected

To be able to exploit the use of Dynamic Network variables you need to have the Networking option enabled on the dongle and both computers configured so that they can speak and see each other on the Ethernet network with the TCP-IP protocol. For further information please consult the section about "NetWorking".

## Dynamic OPC Variables

The Dynamic OPC variables are used for creating dynamic connections with a local or remote OPC Server. The selection of a Dynamic OPC Variable can be easily done by using the **"OPC"** tab in the window which appears after double-clicking on the Variable selection box from any Movicon component or resource.

In order to directly select a Dynamic OPC variable you need to access the OPC Server installed on the local or remote machine. If the OPC Server is configured to startup automatically upon Client request, it is not necessary to manually start the OPC server on the local or remote computer.



You can also specify the name of the Server or the TCP-IP address in the **"Server"** edit box as shown above.

The **"Live Data"**, when enabled, allows you to see the values of the variables in real-time. Once checked, the values are displayed upon the next request.

The **"Refresh"** key functions differently according to the element highlighted in the window's left pane:

- When the local computer or a network computer is highlighted, the **"Refresh"** button executes a refresh of the list of OPC Servers installed on that specific computer.
- When a domain or a network group is highlighted, the **"Refresh"** button executes a refresh of the list of computers available for the one highlighted.
- When a device or an OPC Server group is highlighted, the **"Refresh"** button updates the list of items displayed in the right pane by applying the filter. The filter consists of alphanumeric characters which can be put before or after the "\*" wildcard character. Filters may be for example: My\*, \*Tag\*, \*Tag1

If you cannot access the OPC Server you can directly write the correct syntax in the insertion box as follows:

**[OPC] ServerName.Version\DeviceName.GroupName.TagName**

Where:

**[OPC]** = Movicon Suffix which identifies an OPC connection.

**ServerName** = Name with which the OPC Server is registered in the operating system.

**DeviceName** = Name of device configured in the Server.

**GroupName** = Name of group which variable belongs to.

**TagName** = Name of tag configured in the Server.

In order to use these Dynamic OPC variables you need to have the OPC Client option enabled on the dongle and have an OPC Server already installed and configured. In

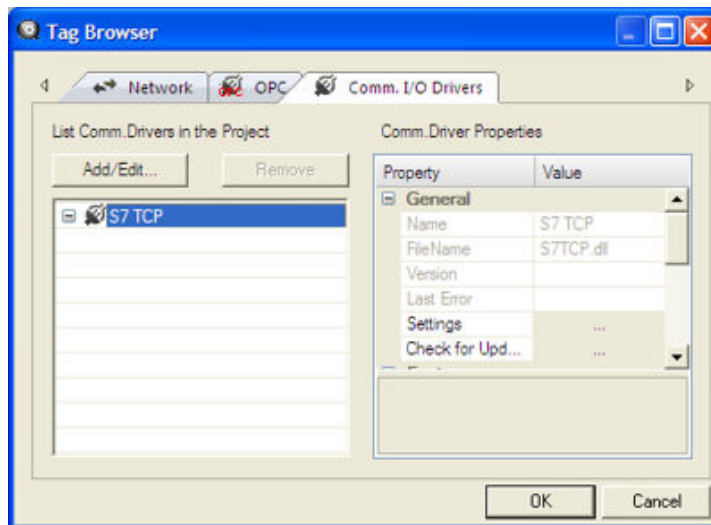


addition, when accessing a remote OPC Server (on another computer) you need to configure the DCOM components appropriately on both the operating systems to get an adequate access level.

## Dynamic Communication Driver Variables

The Dynamic Communication Driver variables are used for creating dynamic links to field devices (ie. PLC) by using the Movicon Communication Drivers. Selecting a Dynamic Communication Driver variable is easily be done through the "**Communication Drivers**" tab from the window which displays following a double-click on the variable selection box of any Movicon component or resource.

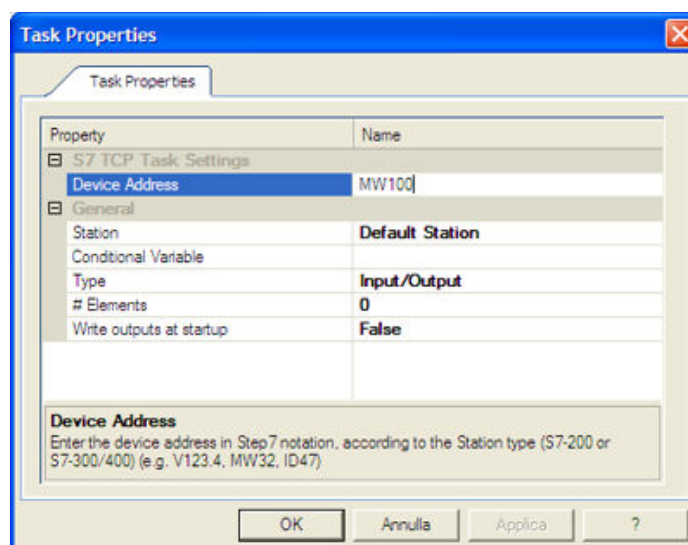
In order to directly select a Dynamic Communication Driver variable, it has to be inserted in the project's "Communication Driver list" beforehand.



The Settings of the selected Driver are displayed on the right hand side of the window.

The driver's configuration mask is opened by using the "**Settings**" button.

By double-clicking on the driver's name displayed on the left hand side of the window or a click on the "**Add...**" button, another window will appear for inserting the communication Task which involves selecting the variable of the device which you wish to link to. The window in question, shown below, shows the different selection fields according to the type of driver being used:



By clicking on the **"Remove"** button it will be possible to delete the selected tag.

If you haven't yet configured the communication driver you can directly write the correct syntax, in the Variable insertion box, as follows:

**[DRV]DriverName.Sta=StationName|Addr=TagAddress**

where:

**[DRV]** = Movicon Suffix which identifies a connection to a Communication Driver

**DriverName** = Name of Communication Driver to be used

**StationName** = Name of station configured in the driver

**TagAddress** = Address of Tag in device (use the syntax of the device itself)

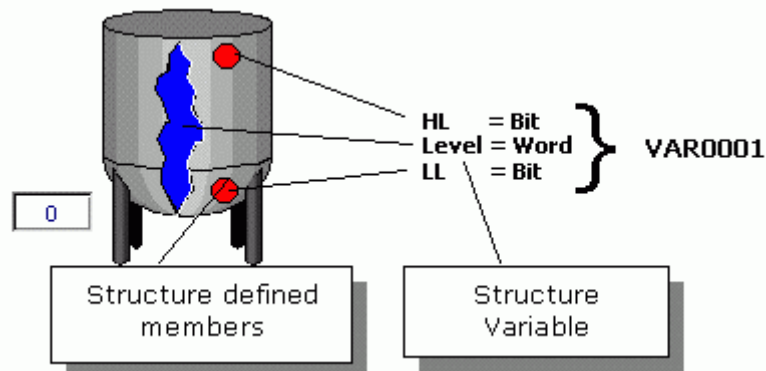
In order to use the Dynamic Communication Driver variables the appropriate communication drivers have to be enabled on the dongle.

## Structure Variables

Movicon provides variables defined as Structures to simplify and slim down the employment of groups, containing a mixture of different type variables, in projects.

These type of variables therefore contain a series of information deriving from other variables of different types, grouped together by one variable known as Structure.

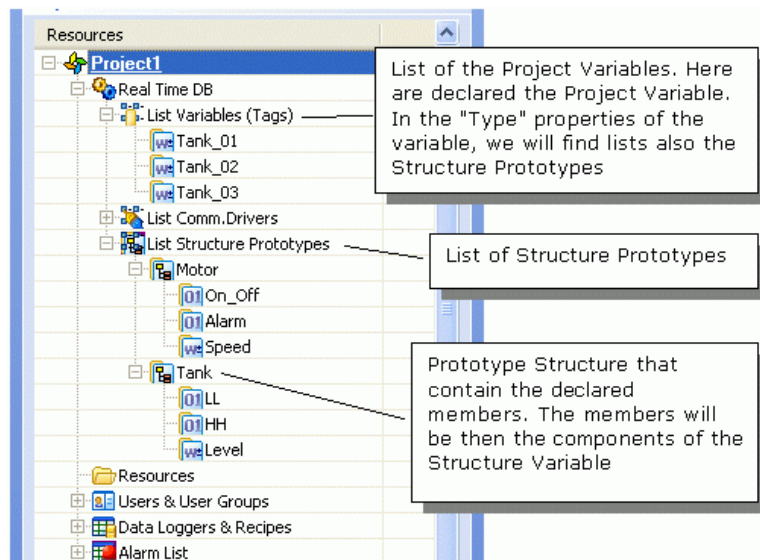
Let's take a tank as an example. Its informative structure is always given by a analogic variable (word type) for the level and two digital variables (bit type) for the minimum and maximum limits. A Structure variable can be composed of three member being a Level (word), HL (bit) and LL (bit). Together they can be called VAR0001, and will appear in the variable types list available for setting variables in the Real Time DB.



Using Structure type variables are very handy when managing repetitive objects made up of groups of data.

Once the Tank object's structure of variables has been setup, it can be inserted into the project many times, each time with a different name (Tank\_1, Tank\_2, Tank\_3, etc.), all referring to the same data structure.

Each variable will refer to its own absolute address, starting from the initial byte, for the necessary number of bytes required by the members belonging to the same structure.



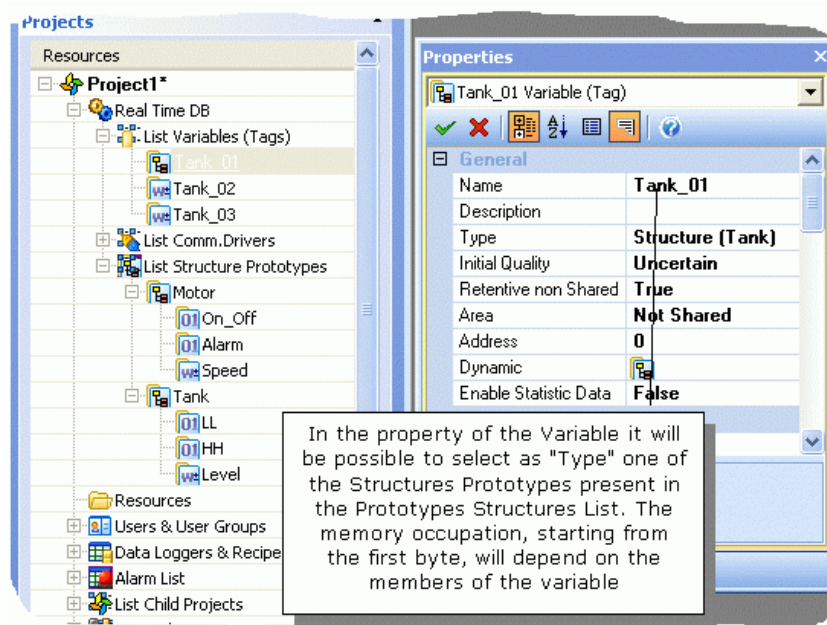
Before inserting a Structure variable into the Real Time DB, its Prototype needs to be set from the project's **'List Structure Prototypes'**.

The Structure Prototypes allow you to setup the Type of structure you wish to create by declaring the Member Variables which will take part in it. After having done this, it will appear as variable **Type** in the Variable properties

It will then be possible to insert new variables as **'Type'** being the name of one of the preset Structure Prototypes instead of bytes, words, floats, etc.

If you select the Input, Output or Flag area in the variable's **'Area'** property, the starting byte address must be specified in the **'Address'** property.

Movicon will always request, as for all the variables in the set shared areas in the project, the start byte address.



The bytes occupied by the structure variables in memory will depend on the types of variable members set in the structure prototype. Any editing of the members belonging to a Structure variable (changing the quantity of bytes) will always cause automatic compacting to take place within that structure variable.



**WARNING!** the "bit" members in structure variables occupy one memory byte. Therefore, if you create a structure prototype containing 8 bit members, the variable will occupy 8 Bytes and not 1 Byte only.

## Structure Prototypes General Properties

---

The general properties are used for associating the main data relating to the Structure Prototypes.

To edit the General Properties of Structure Prototypes, simply select the object with the mouse and use the Movicon **'Properties Window'**.

### Name

The name to be assigned to the Structure Prototype as mnemonic code is entered in this edit box.

Movicon will propose the 'STRUCT' characters followed by a progressive number for default. The default name can be replaced with another name as pleased.



The text for the Structure Prototype's name cannot have gaps or characters which are not alphanumeric. Any reserved gaps or characters can be replaced with the '\_' character.

### Description

A text string to be used as a comment for the Structure Prototype is entered in this edit box.

*The comment will remain associated to the Structure Prototype but will only be available in the database.*

### Size

This field is read only, it reports the live occupation of the Structure Prototype in bytes.

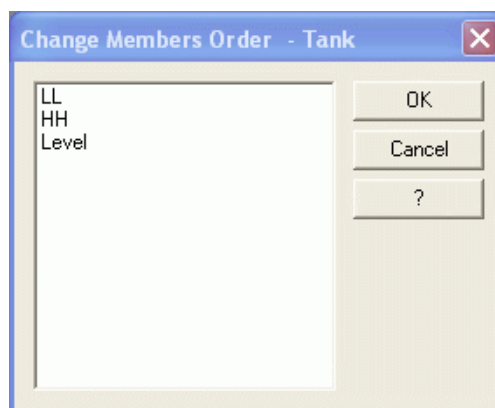
### Member Order

This button allows a window to be opened for changing the order of member variables internal a prototype structure. For further information please refer to the paragraph on **"Variable Member Order"**.

## Variable Member Order

---

The order of variable members of a structure can be edited by using the **'Change Members Order'** command accessed from the **'Commands'** window or right mouse click on the Structure Prototype. When activating the **'Change Members Order'** command the following dialog window will display:



At this point drag the member variable with the mouse to the point required.

## Pointing to Structure Variables

---

When having Structure type variables in the project it will be possible to go and point to each single member by appropriately addressing the variable member selection for the function or logic concerned. In this case you need to specify the name of the structure variable concerned followed by the name of the member variable 's name to be accessed in the syntax. The name of the structure and the name of the member must be separated by the ':' character.

Let's suppose for example that you want to access a Tank level value (tank number 10 of 20 tanks existing on the plant) to which corresponds to the structure variable Tank10 (in which the prototype has been set with the 'Level' variable word), you need to specify:

**Tank10:Level**

which should be:

**<struct name>:<member name>**

Accessing structure variable members can be done through out the project, by assigning this syntax whether to the graphic, Alarms, Data Loggers or logic functions.

If the structure variables are being used within Basic Scripts make sure the ':' is replaced with the '\_' or '.' character when accessing variables directly, without using the Movicon functions (GetVariableValue(), SetVariableValue(), etc). In addition to this you can use the ':' character by enclosing the variable between square brackets. The syntaxes entered are:

```
Sub Main()  
    Dim nVar as Integer  
  
    nVar = Tank10_Level  
    nVar = Tank10.Level  
    nVar = [Tank10:Level]  
    nVar = GetVariableValue("Tank10:Level")  
End Sub
```

Movicon also accepts the <:><member name> syntax for the graphic functions of symbols. However, in this case, you have to specify which is the Default Structure Variable for the symbol as described in the 'Default Structures in symbols' document.

## Default Structures in symbols

---

The functions for assigning Default Variable Structures can be used for each composed graphic symbol.

Assigning default variables offers the advantage to set only one structure of members in the symbol using Default Structure each time. In this way the composed symbol could effectively be independent from the variable, which is set only in the screen's editing phase.

The symbol must be preset with the interested variable member's name, preceded by the ':' separation character, when configuring the graphic animation functions.

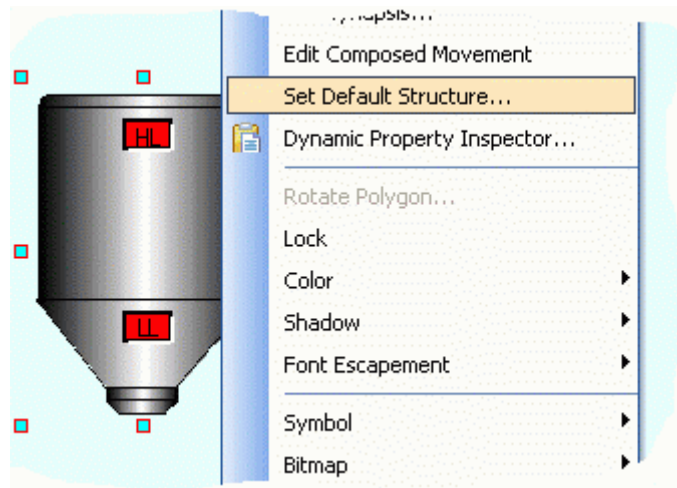
For example, the level representation of the Tank symbol could have the following variable set in the filling function:

**:Level**

The starting Structure variable can be set using the symbol's Default Structure.

To assign the Default Structure you have to select the symbol and use the right mouse key and access the '**Set Default Structure**' command. The selection window will display showing the Variable Structure along with those existing in the Variables DB.

To undo the Default Structure, select the symbol and use the right mouse key, then access the '**Set Default Structure**' command keeping the CTRL key pressed at the same time.

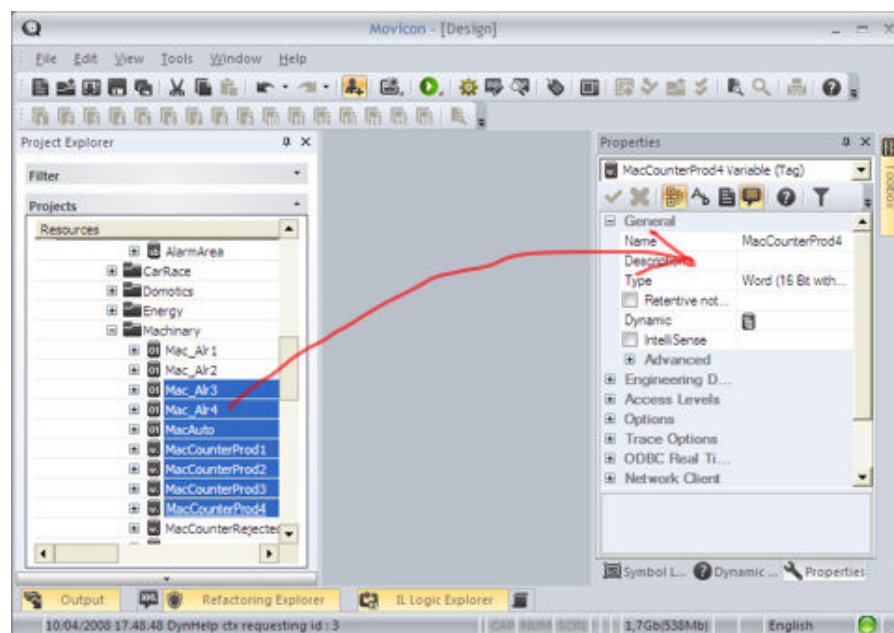


## Variable Properties

The properties of Real Time DB variables are used to determine the variable's starting data area, the association of any commands or association with remote systems or database files.

To set or modify Movicon Real Time DB variables you need to use the **'Properties Window'**. The property window, which can be activated with the standard procedures, displays the variable's property when it is selected from the 'Variables List (Tags)' resource from the 'Project Explorer' window.

Movicon allows you to carry out multiple variable selections for which only the properties common to all the variables selected will be enabled in the **'Properties Window'**. Any modifications carried out in anyone of these properties will effect all the variables selected.



As you can see in the figure above, by executing a multiple variable selection in the 'Properties Window' only some of the properties remain available being those common to all the variables. Any changes to the selected variables can be reported by modifying their properties altogether at the same time.

## General Variable Properties

---

You can associate the main data relating to the Movicon variables by using the general properties.

To modify the General properties of variables select the object with the mouse and use the Movicon 'Properties Window'.

### Name

This edit box is used for entering the name to be assigned as an mnemonic code relating to the variable. The variable's name must be unique as it is used for identifying the variable anywhere in the project.

Movicon proposes the chars VAR followed by a progressive number for default. The default name can be replaced by another name as pleased.



The text for the variable's name must not have any spaces or characters that are not alphanumeric. Any reserved spaces or characters can be replaced with the '\_' character.

### Description

This edit box allows you to type a text string which can be used as a comment to describe the variable within the Real Time DB.



This description will remain associated to the variable but it will only be available inside the Real Time DB.

### Type

This selection is used to specify the type of data represented by the variable in the Real Time DB. The list box will propose the following data types:

Bit  
Byte with sign / Byte without sign (8 bit)  
Word with sign / Word without sign (16 bit)  
Doubleword with sign / Doubleword without sign (32 bit)  
Float (32 bit single precision)  
Double (long) (64 bit double precision)  
String (each character one byte + termination character 0)  
Byte array (fixed length with starting address and byte number)  
Structure: A list of Structure types available will be listed if present on the 'Structure Prototypes' group.



The string type variables can be up to a maximum of 1024 characters.

### Initial Quality

This selection lets you define the variable's initial quality. This setting is significant for those variables which are to be exchanged with other devices or programs, eg. through communication drivers, OPC, Networking. This property is very useful when finding out whether the variables which are read by the field are updated correctly and contain valid values. The variable quality states can be:

- **Good:** valid data
- **Uncertain:** the data has not yet been updated and therefore is not definite
- **Bad:** data is not valid
- **Not Connected:** not connected to field therefore data is not defined

### Retentive not Shared

The variable's retentive is enabled when declared as 'Not Shared' area type.

### Area

This property is used for defining the data area type in which the variable is to be mapped. The options are:

- **Not Shared**
- **Input**
- **Flag**
- **Output**

When you select the 'Not Shared' data area, it will not be necessary to define the variable's absolute address as this will be automatically allocated by Movicon without making any errors of superimposing unwanted addresses. When you use the Input, Output and Flag areas you can associate an absolute address to the variable but special attention must be paid as not to superimpose unwanted addresses.



It is advised to always use the 'Not Shared' areas when possible so as to avoid superimposing unwanted addresses.



**The variables belonging to the 'Not Shared' areas can be exchanged with the field just like the ones belonging to the Input and Output areas.**

#### **Address**

An absolute address must be specified for variables mapped in the Shared (Input, Flag, Output) data areas. The Movicon Shared memory is addressed in bytes. When having to address a bit you must use the 'x.x' syntax (eg. 10.5 = byte 10 bit 5) while the 'x(x)' (Eg. 10(5) = Array of 5 bytes to start from address 10) syntax must be used for byte arrays.



It is advised to always use the 'Not Shared' areas when possible so as to avoid superimposing unwanted addresses.

#### **Auto Alloc**

This command allows you to assign a free address to the variable ("Address" property) within the selected shared area.

#### **Dynamic**

This property is used for associating the variable to a network item, an OPC Server tag or a communication task, directly. You can type the command line, using the appropriate syntax, directly into the appropriated box, or use the 'Tag Browser' window which appears when double clicking the mouse on the edit box.

The syntax which will be reported in the property's box is as follows:

#### **Networking Syntaxes**

*[NET]\\ComputerName\VariableName*

where:

[NET] = Movicon suffix which identifies a network connection

ComputerName = Name of remote computer Server

VariableName = Name of variable in Server project to be connected

#### **OPC Syntaxes**

*[OPC] ServerName.Version\DeviceName.GroupName.TagName*

where:

[OPC] = Movicon Suffix identifies a OPC connection

ServerName = Name with which the OPC Server registered in the operating system

DeviceName = Name of device configured in the Server

GroupName = Name of the tag's starting group.

TagName = Name of tag configured in the Server

#### **Driver Syntaxes**

*[DRV]DriverName.Sta=StationName|Addr=TagAddress*

where:

[DRV] = Movicon Suffix identifies a connection to a Communication Driver



DriverName = Name of Communication Driver to be used  
StationName = Name of station configured in the driver  
TagAddress = Address of Tag in the device (use the device's syntax)

### OPC Group Name

The name of the OPC Group to which the variable is to belong is entered in this field. This setting only has meaning when the OPC link has been inserted in the variable's "Dynamic" property. When no Group name is entered Movicon will insert the dynamic tag in a generic group purposely created for dynamic tags.

### Enable Statistic Data

Enable the statistic data for the variables in question. In this case you will get minimum, maximum and average variable value availability plus other information. To access this information you need to use the variable's appropriate Basic Script functions, such as "DBVarObjCmdTarget.StatisticDataMaxValue", "DBVarObjCmdTarget.StatisticDataMinValue", etc.. Statistic data can be moved from one variable to another by using the appropriate commands in the "Variable Commands List". The variable statistic data, only if retentive, is persistent. The variable's statistic data is then save in the variable's retentive file together with other already existing data (values, quality and timestamp etc.).

### IntelliSense

When enabled, this property consents you to add the variable to the VBA's 'IntelliSense', so that it becomes available in the popup list that appears when pressing the "Ctrl+Space" keys inside the basic script code (also see the section on "IntelliSense use for RealTimeDB variables").

## Variable Engineering Data Properties

---

By using the variable's Engineering data you can execute the variable's value scale. In this way the variables coming from the field are read and the scaled value is written directly in the variable.

To modify Variable Engineering Data Property select the object with the mouse and use the Movicon **'Properties Window'**.



The variable scaling is linear type only and is only significant for variables exchanged with the field. To execute scalings between variables within the project you will need to use the appropriate 'Scaling Objects List' resource which can be accessed through the 'Project Explorer' window.

### Initial Value

This property allows you to specify the initial value to be used for a "Not Retentive" variable. At project startup the variable is set with its initial value before the project's logic is run. A retentive variable (only when in a not shared area) is not initialized on the initial value if there is a valid value in its retentivity file.

Please note that this property is not supported in the basic script interface.

The initial quality value, if used, is set by using the following the rules:

- Integer numeric variables: any value can be set, even those preceded with the +/- signs. The positive number will be considered if a sign is omitted.
- Numeric variables with floating point:: numbers need to be set using the same decimal point configured in the Windows international settings, as separator.
- String variables: any ASCII or Unicode character can be set. The project will need to be set as "unicode" when using unicode characters so that they will get saved.
- Byte array variables: when setting the values of each single byte you will need to separate them with a comma and a space like this: `var_array(3) = "125, 256, 34"`.

- **Structure Variables** : the values of each single byte will need to be set by separating them with a comma and a space like this: `var_structure(1 byte + 1 word) = "125, 256, 34"`.

#### Default Format

This allows you to enter the format preset for default for displaying variables by choosing the various standard value options: "x", "xx", "x.x", etc.

The variable format set will be used in displaying data in the following objects:

- In the **display object** set for displaying variables and not set with a different format in its properties
- In the **numeric pad** to format values displayed in edit boxes. The numeric pad can also manage formats with integer numbers with floating points. Therefore, integer variables can be used for displaying and setting values with floating points, as with the display object, by using a format that includes decimal figures (ie. "x.xx"). If the numeric pad is used with variables without formats, it will behave by default using its base "%.g" format
- **Trend and Data Analysis objects**: are displayed according to the pen variable value format and, as a consequence, the pens scales as well.
- **Meter objects**: are displayed with scales formatted according to the variable format
- **Alarm Window Object**: the variable's format is used for defining the display value format of the variable inserted in threshold's alarm text when using the "%(VAR00001)" syntax. In cases where nothing is specified in the variable's "Default Format" property, the "%f" default value will be used.



**Caution:** Display formats in numeric pads created on screen (see template library), are in "%G" so that variables can be displayed in the most compact form possible. In any case, values with floating points in integer numbers can be set using a preset variable format which included decimal figures.



**Note:** This function can also solve the problems of variables with floating point approximations (floating points in 32 bit) when displayed in numeric pads. The numeric pad will round off numbers with decimal points, as set in the variable format.

#### Engineering Units

By using this property you can associate the engineering units to the variable. This will be displayed in the controls provided for this property's management.

#### Close bit String

When the variable is Bit type, it can be associated with a text string to identify the variable's Close value, being when the bit assumes a value other than zero. In this way, by displaying the variable in the **"Watch Window"** the string will appear directly, replacing the binary value. In addition to this, if the variable has been inserted in a Datalogger, the recording of the variable will show the string's caption instead of the value "1". The DataBase field, for bit type variables, is in fact defined as string type.

#### Open Bit String

When the variable is Bit type it can be associated with a text string to identify the variable's Open value, being when the bit assumes the zero value. other than zero. In this way, by displaying the variable in the **"Watch Window"** the string will appear directly in the place of the binary value. In addition to this, if the variable has been inserted in a Datalogger, the recording of the variable will report the string's caption instead of the value "0". The DataBase field, for bit type variables, is in fact defined as string type.

#### Enable

This setting enables or disables the variables scaling property.

#### Inverse

By enabling this property the scaling will be executed in reverse. For instance if the following was set:

Raw Max. Value = 100  
Raw Min. Value = 0

Scale Max. Value = 1000  
Scale Min. Value = 0

When the variable obtains the raw 0 value, the scaled value will be 1000 and when the variable obtains the 100 value the scaled value will be 0.

**Dead Band**

This property is used for specifying the value to which the scaled data will be set the moment the raw value of the variable should exceed the maximum or minimum set limits. This property is set at '-1' for default.

**Raw Max.**

The maximum unscaled value that the variable can obtain.

**Raw Min.**

The minimum unscaled Value that the variable can obtain.

**Scale Max.**

The maximum scaled value that the variable can obtain.

**Scale Min.**

The minimum scaled Value that the variable can obtain.

**Enable Factor**

This property allows you to use the Gain and Offset factors for scale calculations. The expressions become:

$$(\text{Value} * \text{Gain}) + \text{Offset}$$

You must consider that in this formula the "Value" parameter refers to the Movicon variable and not the Communication Driver's value. Therefore the most explicative formula would be:

$$\text{DriverValue} = (\text{Movicon Value} * \text{Gain}) + \text{Offset}$$

**Inverse Factor**

This property allows you to use Gain and Offset factors in inverse mode for scale calculations. The expressions become:

$$(\text{Value} - \text{Offset}) / \text{Gain}$$

You must consider that in this formula the "-value" parameter refers to the Movicon variable and not the Communication Driver's value. Therefore the most explicative formula would be:

$$\text{DriverValue} = (\text{Movicon Value} - \text{Offset}) / \text{Gain}$$

**Gain**

The Gain value for the scale calculation with the use of multiplication factors.

**Offset**

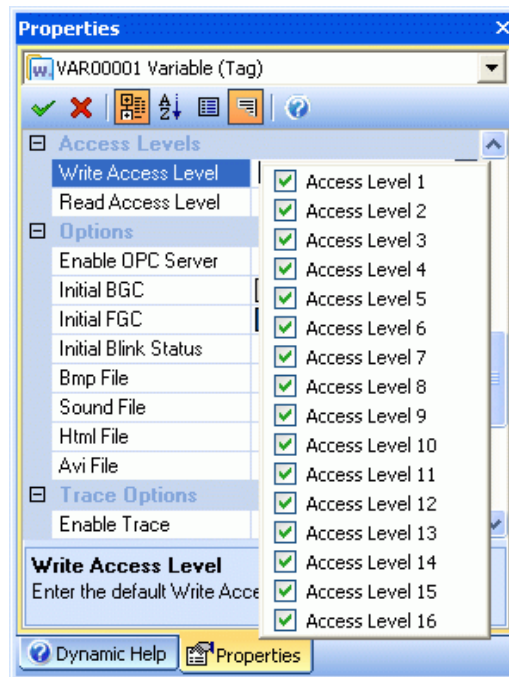
The Offset value for the scale calculation with the use of multiplication factors.

## Variable Access Level Properties

You can enable default access levels in write or in read for the selected variable by using this Access Level property.

To modify the Access Level property select the object with the mouse and use the Movicon **'Properties Window'**.

There are 16 access levels and they can be selected singularly, one by one.



### Write Access Level

The Access Levels can be set through this property for the selected variable in write. In addition to this, the access level of a variable in write is also considered for the users management in the project. A variable can be written:

1. Always when the password management is disabled
2. Always when the "Write Access Level" property is set with the "0000" value (default value)
3. In situations where points 1 and 2 do not count, only when the user's "Access Level" mask encounters the variable's "Write Access Level" mask

In cases where variables can be selected from the "Scheduler window" with this property set for viewing the scheduled plan in "Grid" mode, the Access Level will render variable availability in the selection list according to the logged in user's Access Level.

For further information on 'Access Levels' please refer to the **"Write Access Levels"** paragraph.

### Read Access Level

The Access Level can be defined through this property for the selected variable in read.

For further information on 'Access levels' please refer to the **"Read Access Levels"** paragraph.

### Always Visible

When enabled, this property will render the variable always visible, therefore available for selecting for those environments which require it.

This option is usually used in the following environments:

- In the "Scheduler Window" to determine which variables are to be made visible in the Variable list and therefore selected for commands (only for "Scheduler Windows" in Grid mode)

- When using the Network Server Tag browser, for instance for setting dynamic addresses, where variables are to be always visible and selectable even when the user connected to the Server does not have read access rights

When this option is not set (disabled for default), the variable will only be made visible and selectable in Runtime if the project has the user management active, and the user wishing to access has an access level compatible with the that set for the variable.

For further details on "Access Levels" please refer to the paragraph on "User Levels and Access Levels".

## Variable Options Properties

---

Through this Variable Options Property you can enable the variable to be swapped via OPC and via Movicon Networking.

To modify the Variable's Options properties, select the object with the mouse and use the Movicon **'Properties Window'**.

### Enable OPC Server

Enabling this property will make this variable available to other OPC Client applications.

To connect variables through the OPC Server to OPC Client applications, after having first configured the Movicon OPC Server settings, you need to configure the variables to be connected when enabling this property. Therefore variables will be defined causing a tag to be created.

The Tag is a connection to the Movicon variable made available to OPC Clients.

This operation is not absolutely necessary as tags can also be dynamically defined by a OPC Client by previously selecting the corresponding enabling in the Movicon OPC settings. However all the tags defined through enabling the 'Enabling OPC Server' property are displayed in a item list in the OPC Client and therefore can be viewed and quickly selected.



Whenever you wish to use the OPC Server functionalities, you will have to enable and specify the OPC Server functionality from the project's "Real Time DB OPC Server Settings". How use the OPC functions are described in the respective section.

### Enable Network Server

When enabling this property the variable is made available for sharing with other Movicon applications through the Networking functionalities.

### Initial BGC

Through this property you can associate an initial background colour to the Variable. This property can be interpreted by the OPC Client if predisposed with the necessary functionalities.

For further information please refer to the paragraph on **'Threshold Colour Settings in Drawings and Controls'**.

### Initial FGC

Through this property you can associate an initial border colour to the Variable. This property can be interpreted by the OPC Client if predisposed with the necessary functionalities.

For further information please refer to the paragraph on **'Threshold Colour Settings in Drawings and Controls'**.

### Initial Blink Status

By means of this property you can associate the initial blink status to the Variable. This property can be interpreted by the connected OPC Client if predisposed with the necessary functionalities.

### Bmp File

By means of this property you can associate a Bitmap file to the Variable. This property can be interpreted by the connected OPC Client if predisposed with the necessary functionalities.

### Sound File

By means of this property you can associate a Html File to the Variable. This property can be interpreted by the connected OPC Client if predisposed with the necessary functionalities. This property can also be used for executing a customized sound to the eventual alarm associated to the variable. For further information on this please refer to the paragraph titled "Alarm Threshold Style Properties".

### Html File

By means of this property you can associated a Html file to the Variable. This property can be interpreted by the connected OPC Client if predisposed with the necessary functionalities.

### AVI File

By means of this property you can associate an .AVI file to the Variable. This property can be interpreted by the connected OPC Client if predisposed with the necessary functionalities.

## Variable Trace Options Proprieties

---

You can record on database all the Movicon variable value changes by means of using the Trace properties.

To modify the Trace Options Properties of a variable. select the object with the mouse and use the Movicon **'Properties Window'**.



If an ODBC link is used, the database containing variable Trace information will be created automatically by Movicon in the project's DATAfolder with the name "NameProject\_TraceDB.mdb" in Access2000 format. The file name and ODBC link can be customized through the "Real Time DB Trace DB Settings" property from the 'Real Time DB' resource. A table will be created within the database for each variable enabled with the Tracer.

If an IMDB link is used, will be created automatically by Movicon in the project's 'LOGS' folder a file ".dat" and ".xml" for each variable enabled with the Tracer.

### Enable Trace

This property enables or disables the trace management of the selected variables.



If the trace is enabled for a structure variable, Movicon will create one table only with the Structure Variable's name. However, the names of the Member variables which underwent changes will also then be inserted in the "Changer Column" field.

### Table Name

By using this property you can define the name of the table where the trace data for the variables selected is to be recorded. If this field is left empty the table's name will be the same as the variable's name.

### Data Max.Age

This field is used for defining the how long the trace data is to be stored before being recycled. The time entered should be based on how frequently the variable changes to avoid creating tables with too much data. Practically speaking, more frequently the variable changes, more the recording time should be reduced.



The maximum recording time is to be inserted according to your requirements, keeping in mind the frequency of variable changes and the kind of database to be used. For instance when using a Access2000 database you will more limited with the recording data quantity than you would be with a SQL Server database.

### Enable Day Timeframe

This setting is used for enabling a recording day timeframe when the variable's trace functionality has been enabled. In this case the variable's trace recording will be executed only within the time specified in the **'Timeframe from'** and **'Timeframe to'** properties.

### Timeframe from

The time in which the variable's trace starts is set here when the '**Enable Day Timeframe**' option has been enabled.

### Timeframe to

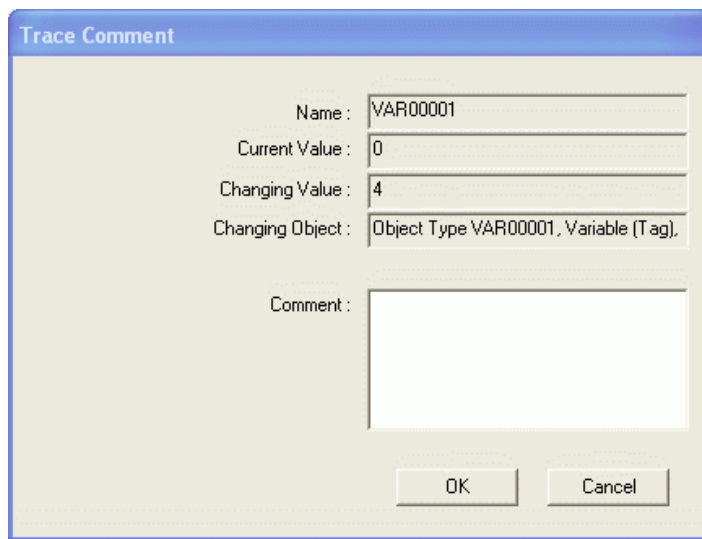
The time in which the variable's trace ends is set here when the '**Enable Day Timeframe**' option has been enabled.

### Add Msg to SysLog

When this setting is enabled a message will be recorded on the 'SysLog' (System Messages' Historical log) every time the variable's value changes. This setting is only valid if the variable's trace is enabled. Also information reported in the database will be recorded in the Historical Log file.

### Trace Comment

This setting, when enabled, allows you to insert a comment in the variable's trace Database each time the variable under goes any changes. This comment will be recorded in the "**Action**" field replacing the text written by Movicon for default. The following will open each time a variable undergoes a change:



At this point the user can insert a comment in the appropriate window and confirm with "OK". The variable's value will only change when the user has confirmed with the "OK" key. If the "Cancel" button is pressed, the variable will not be changed and will keep its previous value.

You should take into consideration that comment do not need to be entered for those changes made to variables from logic which are not subjected to events undertaken by users. For example, this category includes the project or screen IL Logic, Communication Drivers, and the OPC. The window for entering comments will be called if the variable is changed by: Basic Script codes, controls taking action on variables, variable setting commands and in any point of the project they are set.



When the 'Trace Comment Window ' is opened on screen, the variable's value is frozen. Any other process, such as Communication Drivers, IL Logic and Basic Scripts cannot be change the variable's value.

When the "**Add Msg to SysLog**" property is enabled, the comment will also end up in the "DescCol" column from the "SysMsgs" table.



You can customize dialog window's character's font and size by using the appropriate registry keys:

*TraceCommentFont*

*TraceCommentFontSize*

Modifying the font or its size will also change the dialog window's sizes.

### Create DB Table

This command executes the creation of the Variable Trace table within the database. If a table already exists it will be cancelled along with its next recreation when this command is executed. This means that any previously recorded data will be lost.

## Variable ODBC Real Time I/O Link Properties

---

By using the ODBC Real Time I/O Link properties of a variable you can share a variable with a Database for the purpose of making the variable's value also available to other applications capable of reading/writing Database files. The link between the variable and the Database is carried out through the Windows ODBC driver. The ODBC link can be created or modified through the **"ODBC Real Time I/O Link Settings"**.

To modify the Real Time DBMS properties of a variable you have select the object with the mouse and use the Movicon **'Properties Window'**.

### Enable ODBC Sharing

This property allows you to enable the variable which is to be shared with the Database enabled for this purpose. For further information please refer to the section on "Variable Sharing with ODBC".

### Update Quality

By enabling this property the variable's quality state will be updated according to the status of the ODBC link. If any errors are generated by the variable's ODBC link the quality status will change to 'Not Good'.

### Reading Refresh Time

This setting expresses the time (in milliseconds) with which the reading refresh of the variables from the Database takes place.

### I/O Mode

The operating modes of the ODBC link for the specified variable is set in this option box. The operating modes are:

- **read:** means that the variables on the Movicon side are read with a preset frequency time ("Reading Refresh Time"). If this data DOES NOT exist on database the value will not be modified in the application, and if the variable is retentive, it will keep its last value. However the variable's 'quality' can be managed to get information on the value type (when the quality is 'good' this means that the value is the one read from the database, otherwise means that a value has not been set for the variable on database)
- **write:** means on each variation of the variable's value, Movicon will insert the data on database. If the record exists it will be updated otherwise the data will be inserted
- **read-write:** Movicon keeps the variable of the Movicon project and the relative field of the linked Database file at the same value. Any variations of one of the two will consequently cause the other one to change, whether to the project locally or in the Database file. In any case, Movicon will NOT write anything at the Startup the data will be written when the first variation takes place in order to be more flexible

## Variable Network Client Properties

---

This variable 'Network Client' properties allow you to establish a connection between one project's Real Time DB variable to another Movicon project's variable existing on a network station, by using standard protocols such as TCP/IP.

To remote connect in Networking means the dynamic connection between variables contained in the Real Time DB of two or more Movicon PC stations.

The remote connection functions (Networking) are enabled in Runtime only when the appropriate option is active on both the hardware keys (PC Client and PC Server).

To modify the Network Client Properties of a variable, select the object with the mouse and use the Movicon **'Properties Window'**.





**Caution!** When using structure variables, any string type member variables will not be supported and their values will not be updated. The string type variables can be exchanged in networking only when they are part of a structure variable.

#### Enable

This property allows you to enable the Networking connection between local project variable/s and another Movicon Station set up as Server.

The variables status notifications are managed by Movicon are 'event-driven', which means that only notification of status variations are given to optimize the network use.

The connection can use any network capable of supporting the protocols listed in the 'Transport' property, including RAS modem connections, as hardware support.

#### Update Quality

By enabling this property the variable's quality status will be updated according to the status of the Networking connection. If an error is generated in the Networking connection the quality status will change to 'Not Good'.

#### Network Server

The name or IP address of the connected PC Server station must be entered in this edit box.

#### Backup Network Server

The name of the Server PC Station, to be connected to, or its IP address must be entered in this edit box for cases when the main Server is not available. The IP address of any secondary network card of the main Server may also be entered.

The name or IP address of the Secondary Client is also entered in this field for when the Client has to connect to a Redundancy system (re. "**Redundancy**"). In this case the Network Server will be the Primary Server and the Backup Network Server will be the Secondary Server.

#### Network Server Variable

The mnemonic code of the Server project's variable, to which the selected local variable is to connect to, is entered in this edit box. When using the 'Tag Browser' to select the variable directly, you can only select the variable name from those existing in the local Real Time DB and not those existing in the Server project.



When the 'Network Server Variable' is left empty, Movicon will execute the local variable connection to the Server variable which has the same name. This can only be done if this variable exists in both projects (Server e Client).



When connecting variables of different types, Movicon will carry out data conversions to adapt the variable in read to the type of variable associated. However it is the programmer's responsibility to avoid any data lose generated by the conversion (i.e. passing over from 32 bit in read to 16 bit in write).

#### Mode

The Network connection operating modes for the specified variable are set though this option box:

- **Input:** Movicon reads the specified variable's value from the connected Server's Real Time DB and writes its contents on the variable from the local project's (Client) Real Time DB
- **Output:** Movicon writes the value contained in the variable of the local project's (Client) Real Time DB on the variable of the connected Server's Real Time DB
- **Input/Output:** Movicon keeps the connected variables at the same value. Any variations of one of the two will consequently change the one connected, whether in the Local project (Client) or in the Server project

#### Priority

This property is used for associating a priority level to the connection in question. The values are from 0 to 100. The highest number corresponds to the highest level and therefore a maximum of a 100 priorities can be used. The value inserted for default by

Movicon is '-1', that is by selecting the priority set in the **Client Settings** or in any associated **Client Rules**.

## Associating Alarms to Variables

---

Movicon has a very handy functionality which can be used when the project has many alarms with similar text messages and are of the same type associated to a series of different variables.

A description of this powerful Movicon feature can be found in the section headed "Alarms as Templates".

Associating an Alarm to a Variable (or a group of variables) can now be done by just clicking the right mouse key on the Variable desired in the "Variable List" group in the "Project Explorer" window and by selecting the "Associate an Alarm" command, or by using the same command in the Project Explorer's "Command Window".

## Associating Data Loggers and Recipes to Variables

---

Movicon has a very handy function which you can use when needing to set many variables associated to the same Data Logger in the project.

A description of this powerful Movicon feature can be found in the section headed "Data Loggers as Templates".

The association of a Data Logger to a Variable can be done by right mouse clicking on the Variable in the 'Variables List' group found in the 'Project Explorer' window and selecting the 'Associate a DataLogger/Recipe' command or by using the same command also found in the Project Explorer's 'Commands' window.

The request to associate a Data Logger/Recipe will open a further window containing the list of Data Loggers/Recipes available and which were inserted beforehand;

## Associating Events to Variables

---

By using the Movicon 'Event Object' you can create a list of events, each one to be executed upon the change of its associated variable. A list of commands to be activated can be programmed for each event. However this procedure can be inverted by associating a specific event to a variable. This procedure is very handy when diverse variables have to execute the same list of commands, therefore the same event. In this case it would be much easier and faster to create only one event to which associate the command list desired, then associated this event to the variables of interest.

The association of an Event to a Variable can be done by clicking the right mouse button on the Variable in the 'List Variable' group presented in the 'Project Explorer' window and selecting the 'Associate an Event' command, or by using the same command also found in the 'Project Explorer's 'Commands' window.

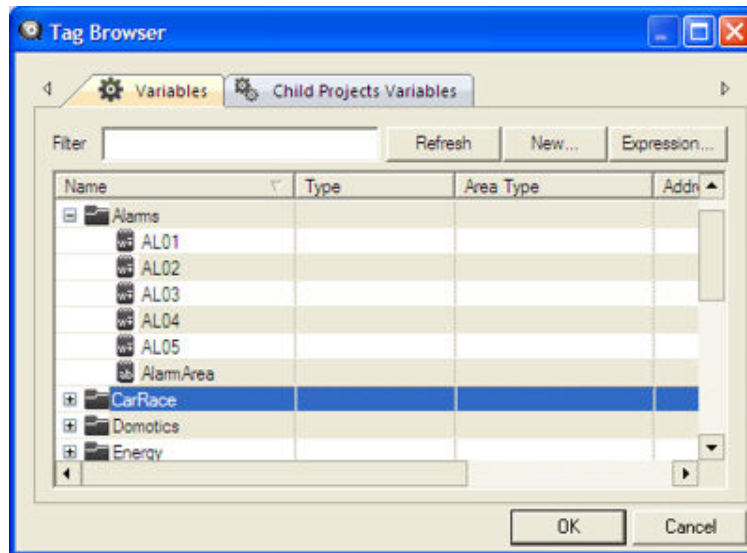
A description of this powerful Movicon feature can be found in the section headed Events as Templates.

# Grouping Variables

Variables can be put into one or more groups within the "Real Time DB" resource. The **"New Variable Group..."** command can be accessed with a right mouse click on the resource or from the command list. The created groups cannot be renamed or cancelled (they do not have properties) from the "Real Time DB" window. They are automatically deleted when they do not contain any variables but only disappear from the "Real Time DB" when the project is reopened in development mode.

Since the order by and filters are applied according to the resource type selected only, the use of the variables groups is useful for speeding up these operations in projects with a very high number of variables.

Groups which are created in this way are also made available from the Movicon "Tag Browser" window.



The "Group" property is found in the "DBVarObjCmdTarget" Basic Script interface and allows you to retrieve the group which the variable belongs to.

## Pointing to a Variable's Bit

In certain cases you may need to "read/write" one single bit of a variable (byte, word, etc). Movicon allows you to execute this operation by using the "VariableName.numbit" syntax. For instance if you want to probe the 0 bit of the "VAR00001" variable declared as Word type, you must write:

VAR00001.0

This type of syntax is supported in almost all the project's resources. You can use it in the following environments:

- Symbol animation properties
- alarms
- IL Logic
- Event Object list
- Scheduler Object List
- Generally in all those resource fields where a variable can be inserted



**Variable bit Pointing is NOT supported in the Movicon Basic Scripts.**

## 6. Screens

---

***The Screens are the fundamental resources in creating graphic interfaces. This chapter describes the Screen resources, by also referring to the appropriate chapters on the techniques used for inserting drawings, commands, controls into Screens.***

The Screen window is one of the key elements in a Movicon project. The Screen is used to supervise the process (or part of it) by using the graphic commands for animations activated by field variables.

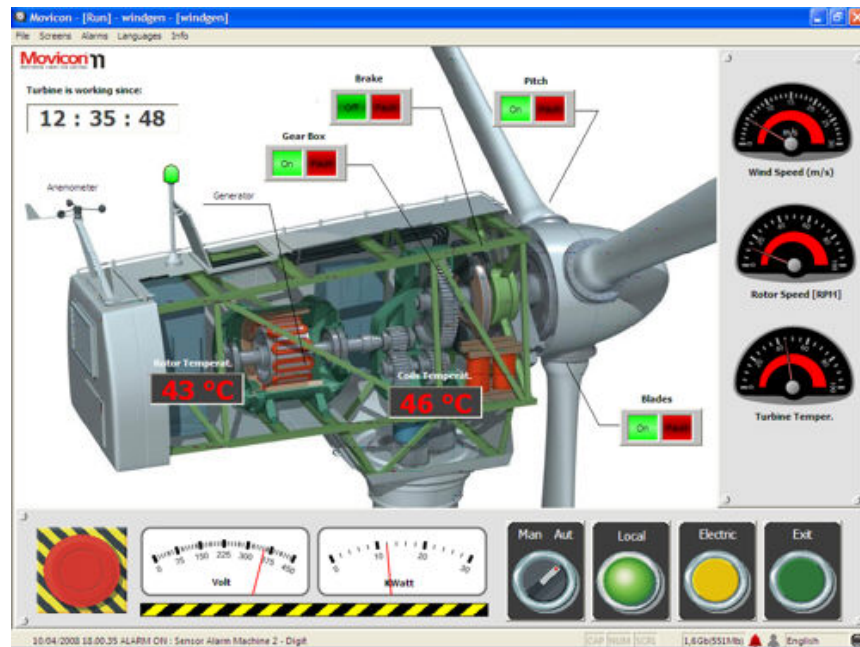
The Screen controls, described in the appropriate chapters, are used for setting commands or variables to the field, as well as facilitating supervision tasks.

The Movicon workspace can be composed of one Screen only, but using the "Embedded View" control is possible display a Screen within other Screen.



The Screen window represents the projects screen pages. However Screen can be inserted into other Screens (by using the 'Embedded View' control) to get a composition of more Screens in the same page. In this case, however, the parent Screen will always remain the Container Screen.

The Screen windows are preset to contain, apart from graphical drawings, controls such as all the graphical command functions or displays already preset by Movicon. The descriptions for inserting and setting controls in Screen windows are found in the appropriate chapters.



An example of the Screen page. A Screen is the container of controls and designs and can receive background image files.

# Inserting Screens

---

When you wish to insert a Screen into the project you need to start this procedure by inserting a new object in the "Screens" group residing in the 'Project Explorer' window. You can insert a new Screen either by right mouse clicking on the "Screens" group in the 'Project Explorer' window and then select the 'New Screen' command or by using the appropriate icon from the Movicon Tool bar. When using the later you have to left mouse click and keep pressed on the icon for about one second to open a drop-down list where you can select the resource to be inserted which should be the Screen resource in this case.

When confirming this operation the new Screen window, with default sizes and positions, in the group or in the point selected in the project structure. At this point you can go on and set the new window's properties as described in the documents about **"Screen Properties"**.

The resource can next be assigned a Name by clicking the resource and entering a name to replace the one proposed for default, or after having selected the resource press the F2 key and insert the new name.

## Importing Screens from other Projects

---

Movicon allows one or more Screens to be copied from one project to another. In order to do this you must first open both projects, then select the Screen from the Project Explorer Window of the source project, execute the Copy command then point to an area in the "Screens" group of the Project Explorer window of the destination project and execute the Paste command. The copied Screens will then also be available in the destination project. You can also use the Drag & Drop technique with the following procedure: select the Screens from the source project and, by keeping the mouse left key pressed, drag them to the point desired in the destination project and then release the mouse button.

# Startup Screen

---

Movicon has been designed to open the Screen at the start of the project runtime which has been selected in the **"Startup Screen"** box in the **'Project Execution Settings'** properties. If these properties are not configured the Movicon project startup will not open a screen page leaving the operator to activate one by using the commands in the project's Menu and Accelerators if previously configured only.

*The Startup Screen can be selected from any of those preset by the programmer, independently from its name.*



Selecting the "Startup Screen" is indispensable so that when the project is put in Run mode a startup page is displayed, which usually represents the plant's general Layout or a simple presentation page.

**When a project is Run from the development environment with a screen still opened in the workspace, this screen will be considered as the startup screen. This is very handy when the screen you are working is started up directly.**

# Opening Screen Modalities

Once created and configured the Movicon Screens can be opened in different ways based on the type of command being used. This allows the programmer to display the Screen in different ways as he/she requires. A Screen can in fact be opened as a simple Video Page, or as a Modal Window acting as a Dialog Window which opens on top of the current page.

As mentioned beforehand, the opening Screen modalities are selected by means of the '**Screen Commands**' settings residing in the '**Command List**' of each Movicon control.

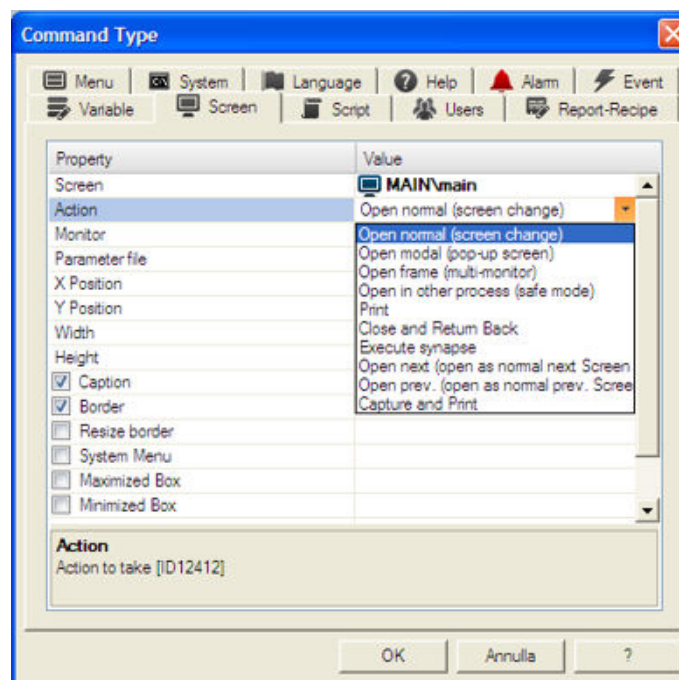
The "**Startup Screen**" is displayed at project startup which will be opened as a normal Video Page.

The screen resources are saved in binary during project Runtime. This makes loading it in memory faster. This is more advantageous in WinCE where the XML parser in the resources is quite slow. In fact, the handling of the XML parser is not of interest when a screen (in compiled form) is opened on screen. The method applied is:

The first time a screen is opened on screen during Runtime, a control is made to see whether the "ScreenName.movscr\_c" file exists and whether the data is the same or more updated than the "ScreenName.movscr" xml file. If this is so, it is then loaded instead of the xml file, otherwise the xml file will be loaded and the "ScreenName.movscr\_c" file will be saved in binary mode. This operation is slow when the screen is opened for the first time but the next will be faster.

This function is managed by the "MoviconRunTime.exe", "MoviconService.exe", "MovXCE.exe" and "Movicon.exe" modules when executed with the /R. option. This management is not activated when the project is run directly from the development environment.

The opening Screens Modalities are as described below. For further information see the paragraph "**Screen Commands**".



## Open Normal (screen change)

The "Open Normal" command opens the Screen as a Video Page. This modality is used for page changes, therefore for passing from one page to another. When the opening of a Screen is called the previous one will be closed and unloaded from the RAM, unless the "**Not destroyable**" option has been enabled in the 'Screen Style Properties'.

### **Open Modal (pop-up screen)**

---

The 'Open Modal' command evokes the opening of a Screen as a Modal window. In this case the Screen is opened acting as a dialog window, therefore on top of the Video Page active at that moment. Nevertheless, in this condition the current page will remain in background and the Modal Screen will remain in foreground. It will not be possible to execute commands outside the Modal Screen area with the mouse until the Modal Screen is closed with the appropriate procedures. The sizes and positions, in which the Screen is to appear with, can be specified in the parameters which are a part of this command.

This functionality is usually used when the operator has to execute the settings or commands etc..

### **Open Frame (multi-monitor)**

---

The 'Open Frame' command evokes the opening of a Screen in a window different from the Movicon Main window. The window in question will remain at the forefront but will not be modal, therefore you will also be able to execute commands presented in the window underneath.

### **Open in other process (Safe Mode)**

---

This command allows the opening and displaying of a Screen executing another instance. (therefore a process in a different memory area). This is useful when Screens using ActiveX might cause an error in the main application.

Using this mode however consumes more overall resources whether for the Screen in 'Safe Mode' (which acts as client) and for the one loading it (which acts as server).

### **Print**

---

This command allows to send to the default printer a print of the selected screen.

### **Close and Return back**

---

This commands force the closing of the specified screen or, if no screen is defined, of the active screen. If the active screen is a modal (pop-up) window, it will be simply closed. If the close command refers to a standard screen, Movicon will go back to the last loaded window.

Movicon manages a list of the previously opened screens. The maximum number of the "back" commands is 10 as a default value, but it can be changed through the "Back History Max Depth" property of the "Screen Navigation Editor " project resource.

### **Open next - Open Prev**

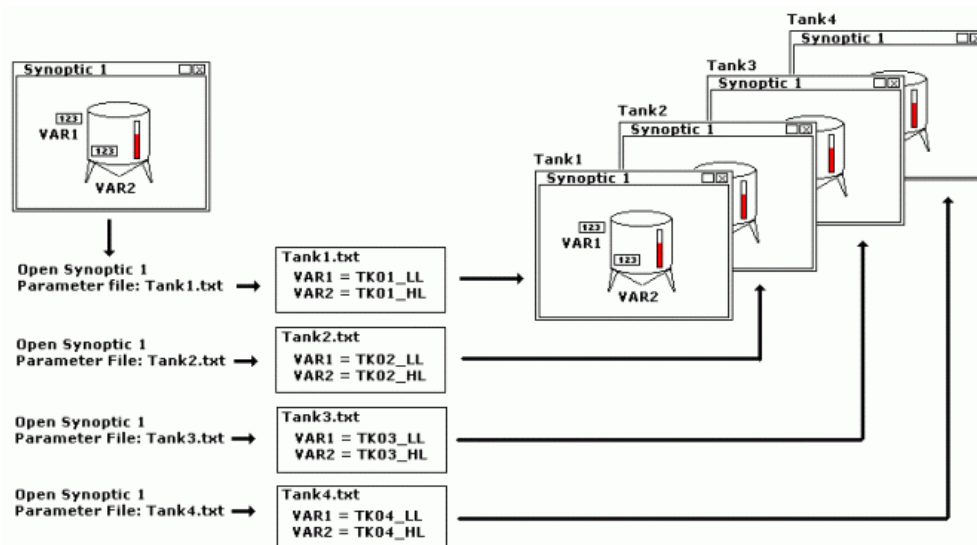
---

This command allows to open project screens using the Screen ID number ("ID" property in "General" screen properties) instead of the screen name. You should properly set a screen ID for each screen (different from 0) and use this function to create a set of "<<BACK "and "NEXT>>" pages navigation commands. See also Screen Change commands based on Screen IDs

# Screens with Parameters

The up-to-date automation techniques often require resources with parameters, in order to use only one resource repeatedly to get the required parameters.

The parameter techniques applied to Screens permit one only screen to be created, which can be called up many times with different parameters. In this case the parameters are the variables from the Movicon Real Time DB.



Movicon allows the functionalities, associated to displaying and animating the screen's vectorial drawings and Templates, to be indexed. The programmer will find this functionality extremely useful when handling projects containing screens with identical graphics, but different variables associated.

Let's take a plant containing 4 identical tanks as an example where only one screen is to be created and which can be called up by 4 buttons, one for each tank. The screen has to have parameters, which means it has to contain dummy variables which will be replaced in Runtime with real variables of each single tank.

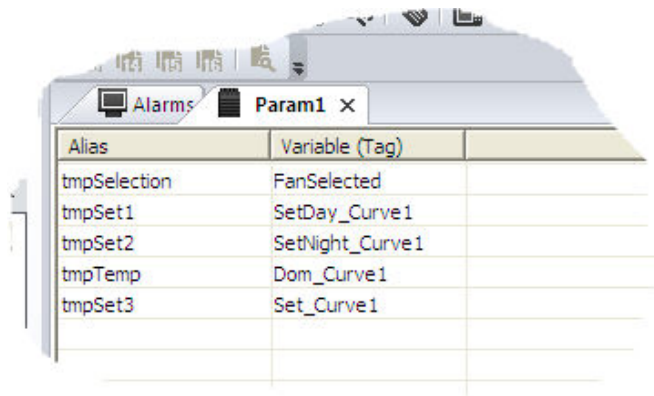
In this case it is clearly necessary to have use of the indexing techniques (or parameter techniques), so that the variables associated to the Screen in the programming stage are replaced by the ones needed during Runtime, in function with the parameter file used for opening the Screen.

The parameter file is a simple text file (UNICODE format) within which the associations between the 'dummy-variable' and real-variable' are specified. This file has to be created by the programmer, and is to be inserted in the **"Parameter File"** properties of the **'Screen Commands'**.

## Parameter File

Movicon provides a resource through which you can edit these parameter files directly from the Movicon development environment. By invoking the "New Parameter File" command from the Project Explorer "Parameter Files" Group you can directly edit a file in table format. The following window will open:



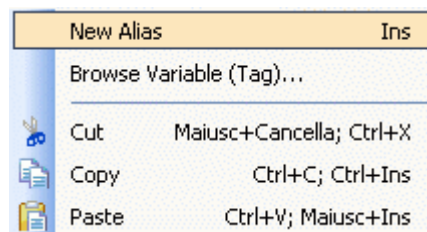


Alias	Variable (Tag)
tmpSelection	FanSelected
tmpSet1	SetDay_Curve1
tmpSet2	SetNight_Curve1
tmpTemp	Dom_Curve1
tmpSet3	Set_Curve1

The Alias column represents the name of the "Parameter Variable", being the name of the variable which will be replaced with the name of the variable to be displayed each time on screen. The "Variable (Tag)" column represents the effective variable which must be passed to the screen to replace the Alias one. The parameter files associated to the same screen contain the same Aliases, while the variables associated to the Aliases are different. The parameter files are created in the project's "RESOURCES" folder by Movicon and will be saved in UNICODE format with the ".movpar" extension. The file's internal will be structured with the following syntax:

<Alias>,<Variable(Tag)>

When right clicking on the table the parameters will appear in a text menu with the commands needed for inserting a new Alias and selecting variables from the Real Time DB:



where:

**New Alias:** adds a new line to the table for inserting a new parameter

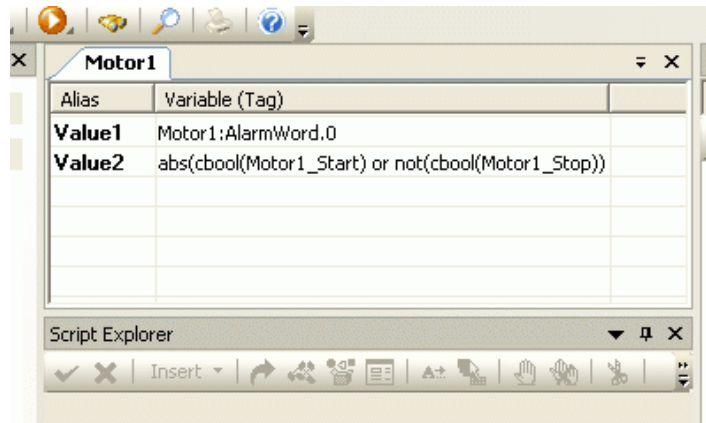
**Browse Variable(Tag)...**: allows you to select the variable from the project's Real Time DB with the Browse window



The browse window, for selecting variables, can be opened directly by clicking on the "Variable (Tag)" field while keeping the ALT key pressed.

The parameter file can always be edited manually with a normal text editor as long as the characteristics described above are respected which involve the syntaxes inside the file and saving it in UNICODE format. The files created by the Movicon resource and those created manually should be structured in the same way.

In the "Variable(Tag)" field you can also insert a sting that doesn't necessarily have to correspond to the name of a variable from the RealTimeDB. You can specify a single bit of a variable or also a basic expression instead which practically means you can enter the syntax which is usually supported in the field where the alias has been inserted:



### Example

In the example described above a Screen has been created with a tank and two animation variables, VAR1 and VAR2 (dummy variables). At this point the parameter file has to be created for each Tank to be displayed where the associations between dummy variables and the real variables are to be done. The four files shown below have been created with a text editor, but their contents are structured in the same way as those created by Movicon.

```
Tank1.txt
VAR1,TK01_LL
VAR2,TK01_HL
```

```
Tank2.txt
VAR1,TK02_LL
VAR2,TK02_HL
```

```
Tank3.txt
VAR1,TK03_LL
VAR2,TK03_HL
```

```
Tank4.txt
VAR1,TK04_LL
VAR2,TK04_HL
```

As you can see the dummy variables are always the same in the four files, while the real variables change according to the Tank. At this point the only thing remaining to do is to associate the parameter file, one at a time, to the required Screen opening command to display the data of one tank or of another.



The opening of a Screen with parameters can also be executed with the appropriate Basic Script functions as well as by using the Movicon **'Command List'**.

## Local Screen Variables

The screens can contain local variables. The local variables are initialized only when the screen is loaded in memory and can be used only inside the context of that screen. Therefore they can be used in the Screen's I Logic and Script codes and in the objects it contains. Keep in mind that:

1. the local variables in the script codes can only be managed by using the variable's name directly. The `GetVariableValue()` and `SetVariableValue()` functions do not support local variables

2. when a local variable has the same name of a variable contained in the Real Time DB, priority will be given to the local variable. To access Real Time DB global variables you must use the following syntax:

..\<variable name>

Caution: this type of syntax is supported in the animation and command properties and IL Logic only. Therefore it is not possible to use the syntax in the Basic Script code.



The "..\" suffix is needed for going back one hierarchy in a variable pointer. The hierarchy is as follows:

Local Screen Variables -> project Variables -> Parent project Variables

for instance, to access a variable from a parent project with the same Local Screen Variable name you must use this syntax:

..\..\<variable name>



The Local Variables cannot be retentive and cannot be used for communication. Furthermore they do not have all the properties that the global variables have.

Local variables are very handy to use with the symbol libraries. These variables are in fact exported to the template library on a par with global variables. When a template is then inserted on screen, its variables will be created as local variables by keeping the SHIFT key pressed down. If the SHIFT key is not pressed down the variables will be created as global variables in the Real Time DB.

Local Variables can be added by selecting the screen from the Project Explorer window and then activating the "Add New Local Variable..." command.

## Numeric and Alphanumeric Screens

---

Movicon allows you to customize Numeric and Alphanumeric Pads which are opened up the "numeric pad" and "alphanumeric pad" commands, or with the respective basic script functions. Customized panels can be created by using the Movicon screens and symbols. The customized panel can be created with a screen within which an "Editable Display" and a series of "Buttons" can be positioned. You can then associate the "Append Value", "Remove Value", "Swap Plus-Minus" and "Append Decimal Mode ON-OFF" commands to these buttons so that the values in the display can be changed. To allow Movicon to manage this screen as a Pad you must insert it into the project's "Numeric Screen" or "Alphanumeric Screen" execution properties. When a request is made to open a Numeric or Alphanumeric Pad, Movicon will open the customized screen instead of the default Pad. The screen will be opened in modal mode. In addition to the two Numeric or Alphanumeric screens some of the system's local variables are managed so that the Pad can work properly. These local variables are:

Numeric Screen:

value  
value\_  
minValue\_  
maxValue\_  
title\_  
OK\_

Alphanumeric Screen:

value  
value\_  
maxChars\_  
title\_  
isPassword\_  
OK\_

The meaning of these variables are as follows:

**value** = current value of the variable to be changed. When Movicon loads the screen, the local variable is set to the same value as the variable to be set through the "numeric pad" and "alphanumeric pad" command. The "value" variable should be declared as string type if an Alphanumeric screen is used, or numeric type (i.e. Double) if a Numeric screen is used.

**value\_** = new value to be assigned to the variable through the "numeric pad" and "alphanumeric pad" command. When Movicon loads the screen, the "value\_" variable is set to zero.

The "value\_" variable should be declared as string type if an Alphanumeric screen is used, or numeric type (i.e. Double) if a Numeric screen is used.

**minValue\_** = the minimum value which the variable may obtain. When Movicon opens the Screen the local minValue\_ variable minValue\_ is set with the same value in the Pad's open command "Min. Value" field. No error message will be generated when this limit is exceeded but when the Screen is closed the value will not be transferred to the variable. The minValue\_ variable must be declared numeric type (eg. Double)

**maxValue\_** = the maximum value that the variable may obtain. When Movicon opens the Screen the local mxValue\_ variable is set with the same value in the Pad's open command's "Max. Value" field. No error message will be generated when this limit is exceeded but when the screen closed the value will not be transferred to the variable. The maxValue\_ variable must be declared numeric type (eg. Double)

**maxChars\_** = maximum number of characters that the variable may obtain. When Movicon opens the Screen the local maxChars\_ variable is set with the same value in the Pad's open command's "Max. Chars" field. No error messages are generated when this limit is exceeded but when the screen closes the value will not be transferred to the variable. The maxChars\_ variable must be declared numeric type (eg. Word)

**title\_** = Pad title. When Movicon opens the Screen the local title\_ variable is set with the name of the variable to be modified. The title\_ variable must be declared string type

**isPassword\_** = this variable is set at 1 when the Alphanumeric screen is called from a password entry window. This allows the display to be set with the "Password" property and therefore display text entries are protected. The isPassword\_ variable must be declared numeric or Bit type.

**OK\_** = this variable is needed in the screen's closing phase. If its value is different from zero, the local value\_ variable's value will be returned to the variable to be modified when the screen is closed. If, however, its value is zero, the value of the variable to be modified will remain unvaried when the screen is closed. The OK\_ variable must be declared numeric or Bit type



Local variables as described above should be created by the designer when needed, respecting the proper syntax. If Templates from the Symbols library are used, when the Template is added to a screen, the local variables will be automatically created.

## Screen Change commands based on Screen IDs

---

You can insert commands for opening screens before or after the current screen open, based on a map determined by the value of the "ID" property of each screen. Two commands, "Open Next (open as normal next ID screen)" and "Open Prev. (open as normal prev. ID screen)" have been provided for this purpose in the "Screen" Group from the Command List.

To insert a screen onto the scrolling map, it will need to have the ID value greater than zero. If the ID is equal to zero, the screen will be excluded from this navigation. Normally the ID value is left at zero for those screens to be excluded from the navigation such as those used inside objects as Embedded Screens or TAB Groups (page footers or headers).

**You must also take into account that these navigation commands do not have effect if they are used when a screen is active with its ID set at zero. In addition to this, screens will open in "Normal" mode only and not in "Modal" or "Frame" mode.**



**These navigation commands are not supported outside the screen context. For instance, they cannot be used from menus, accelerators, event objects, schedulers, etc.**

The screen map is based on the screens' IDs and created by Movicon the first time a command of this type is executed. The screen pages are scrolled in sequence to their ID numbers. If the screen IDs are modified or other screens are added or removed during application runtime, the map will become obsolete and recreated again.

These screen navigation commands are also supported in Web Client and can be extended to child project screens.



When a the screen with the lowest ID number is open, the "Open Prev." command will not do anything. This also goes for the "Open Next" command when the screen with the highest ID number is open.



Screens with the same ID number will be inserted on the list at random only. This condition should be avoided and considered a planning error.

## Screen Resize

---

In design phase, when the size of a screen is changed, setting the "Width" and "Height" properties, Movicon will ask if the objects included in the screen should be resized keeping proportions. The message is the following:

The screen is going to be resized. Do you want to scale up or down all the symbols in this screen?

Answering "Yes" the objects will be proportionally resized and placed in order to adapt to new screen size. Answering "No" the objects will keep their size and position and could be found out of the screen if it has been reduced.

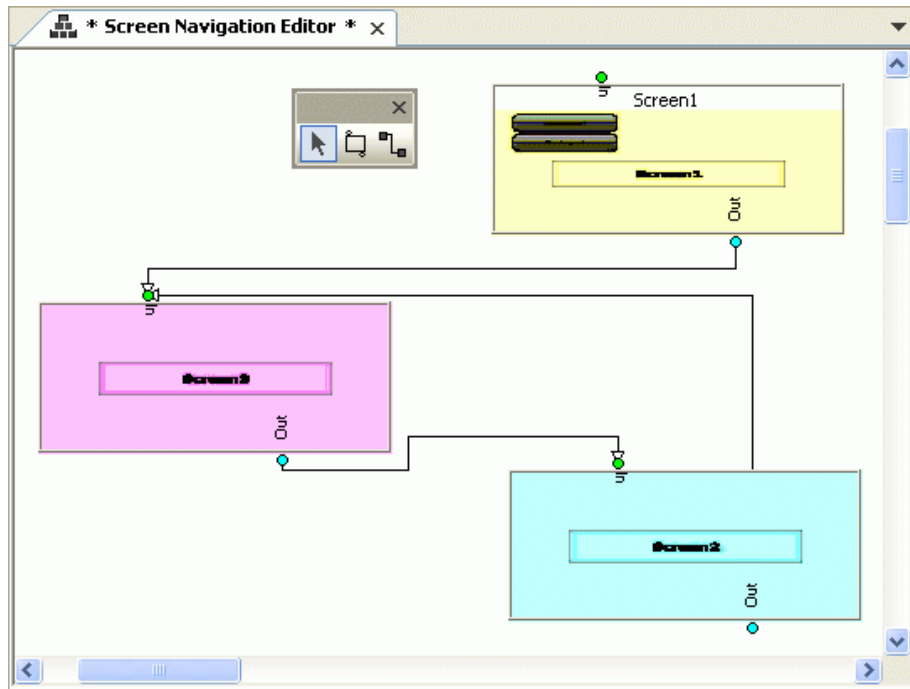
## Screen Navigation Editor

---

***The Screen Navigation Editor resource allows you to manage screen navigation graphically and intuitively.***

By using the "Screen Navigation Editor" resource you can define screen navigation graphically. This means that you can create screen connections that will automatically convert into a navigation bar to display in the different screens during runtime.

This is done by creating a navigation map using an editor specially for this purpose and which is opened with a double-click on the "Screen Navigation Editor" resource from the "Project explorer" window. This editor opens up with an empty screen within which you can create the screen navigation map:



An example of a "Screen Navigation" page.

## Creating the Navigation Map

The "Screen Navigation Editor" resource will open as an empty screen that can be set with a different back color, size etc., through its properties. "Embedded Screens" from the Toolbox are then inserted in this screen by using the appropriate commands. Each embedded screen has two dots known as Synapses one for Input ("In" green dot) and one for Output ("Out", light blue dot). The associations of the Embedded Screens to be displayed are done through their properties. You can create a navigation network by connecting the Embedded Screens to each other. For instance, if you connect the "Screen1" Out Synapses to the "Screen2" In Synapses, "Screen1" will then be able to provide commands, such as an open command, to open "Screen2" during runtime. During the Runtime mode the navigation map is converted into a "Button Navigation Bar" and made available in the screen opened at that moment. In the Button Bar, which will be generated for each screen, a button will be made available for each out connection ("Out" Synapses) presented in the navigation map of that screen. The name of the destination screen will be displayed in each bar button.

Once the "Screen Navigation Editor" window is opened, a toolbox will display with the tools needed for creating the navigation map:



By using this toolbox you can select the following objects:

**Pointer:** this selection consents to restoring the mouse back to its original pointer image in order to select the various objects on screen.

**Screen:** this selection consents to adding an Embedded Screen on the page. The Embedded Screen will already have an "In" and "Out" Synapses and once inserted a window will open for selecting a Screen to associate to the object. The Embedded Screen will display the Screen's static image but you can change the associated Screen through the object's properties later. After having inserted the object the mouse pointer will be automatically restored and therefore you will need to select the command from the toolbox again for inserting another object.

**Connector:** this selection consents to connecting an "Out" Synapses to an "In" Synapses. In order to do this you will need to click on the "Out" Synapses of one Embedded Screen (the pointer will change image only when clicked on the right zone) and the click on the "" Synapses of another Embedded Screen to connect them up. This procedure can then be repeated for all other connections desired. To restore the pointer back to its original image you will need to select the "Pointer" command from the Toolbox.

The navigation layout can be also be displayed in Runtime like any other screen page. In order to do this you will need to insert a "Screen Command" set as a "Open Normal (screen change)" action, and using "\* Screen Navigation \*" as the screen's name. This will allow you to get a graphical layout of the screen navigation map to use in runtime for changing pages: a click on an embedded screen on the map will open the screen it represents directly.



The screen navigation configuration set in the child project is used and not the one set in the Parent project when opening child project screens. Therefore different screen navigations are consented according the child project context being navigated.

You can further customize the navigation button bar with other buttons by using the "Screen Navigation Style Properties" which allows buttons to be added for going back to a previous screen or to open the Startup Screen directly, etc.

## Screen Navigation Style Properties

---

Some of the Screen Navigation bar parameters can be set through their style properties. This is done by selecting and opening, if need be, the Screen Navigation Editor resource to access its properties through the Movicon **Properties Window** and modify them.

### Add Back Button

When enabled, this property consents a button to be added to the navigation bar for returning back to the previous screen.

### Back History Max Depth

This property is used for inserting the maximum number of screen that can be scrolled backwards with the "Back" button (or by using the "Close" command from the Screen Command List without specifying the screen's name).

### Add Back to Startup Button

When enabled, this property consents a button to be added to the screen navigation bar for returning to the main screen set as the project's "Startup Screen".

### Assign Shortcuts

When enabled, this property allows a shortcut to be assigned automatically to the button inserted on the navigation bar. The F1 to F4 keys will be used as the shortcuts in button used for loading new screens, while the "backspace" key will be used for returning to a previous screen and the ESC key for returning to the main screen. These shortcut keys are also displayed in brackets along side the buttons' texts.

### Back Button Text

The text to appear on the navigation bar's **"Back button"** is entered in this editbox. The default text will be used if this field is left blank.

### Startup Screen Button Text

The text to appear on the navigation bar's **"Startup Screen button"** . The default text will be used if this field is left blank.

### Button Size

The **"Button Size"** is described in the Drawing and Control **"Style Objects Properties"** section.

### Align Buttons

The **"Align Buttons"** is described in the Drawing and Control **"Style Objects Properties"** section.

# Screen Properties

---

The Screens inserted into the **"Screens"** folder in the 'Project Explorer' window can be completely customized in their properties. In order to this select and open the Screen required and then edit its settings through the Movicon **'Properties Window'**.

## Screen Animations

---

Movicon manages animations in screens by using the timer notifications. For instance, blinking alarms or blinking object background colors, are linked to this management executed with timers in Movicon. There are two internal Movicon timers: the first timer controls the "fast" animations (blinking alarms, fast blinking backgrounds, etc); the second one controls there other animations (average or slow blinking backgrounds, etc). You can modify this timer management through the screen's "General" properties.

## Screen General Properties

---

The General properties are used for defining the Screen's sizes and the position and sizes of any associated background files. In order to do this just select and open the Screen required and then edit its settings through the Movicon **'Properties Window'**.



When sizing the window on the Web Client side, the screens allocated on the Server side will be sized in the way as requested by the WebClient without controlling any maximum limits. This may be a problem for all those platforms, especially for WinCE where memory is limited, and also in Win32 left open to possible memory saturation on the Server in uncontrolled cases. Therefore, four properties have been provided for each screen on the WebClient side, described below, which consent you to choose the screen's max size on the WebClient side and also the max size of the packs sent by keeping them set at low values so that they are small enough and optimal for reduced bandwidths.

### Name

This edit box is used for entering or modify the Screen name.

### ID

Movicon let's you define a ID number for the Screen window. The ID code can be read in word format in the appropriated 'System Variables' when the Screen is active and for each Screen window can be declared a ID number which can be specified in this edit box. The purpose of this setting is to know which is the active Screen at a certain moment using the Movicon logic by reading the values returned from the 'System Variables'.

### Width

The Screen's width in pixels, which it is to be displayed with, is entered in this edit box.

### Height

The Screen's height in pixels, which it is to be displayed with, is entered in this edit box.

### Max. WebClient Width

This property sets the maximum screen width size when displaying in the WebClient. No sizing limit will be enforced when leaving this value set at zero.

### Max. WebClient Height

This property sets the maximum screen height size when displaying in the WebClient. No sizing limit will be enforced when leaving this value set at zero.

### Max. WebClient Packed Width

This property sets the maximum width size of the frames sent to the WebClient. No sizing limit will be enforced when leaving this value set at zero.



### Max. WebClient Packed Height

This property sets the maximum height size of the frames sent to the WebClient. No sizing limit will be enforced when leaving this value set at zero.

### Fast Timer Tick Animation

This edit box is used for entering the time frequency in milliseconds to double-check the screen's "fast" animations. This value does not change the fast timer tick (default 500 milliseconds), it just changes the frequency with which these animations are double-checked. Therefore, this value obtains a high precision in the timer ticks, regardless of the amount of resources being used.

### Fast Timer Tick Loop

This edit box is used for entering the value which expressed the maximum number of animations managed for each fast timer tick. Each tick is executed with the frequency set in the "Fast Timer Tick". By increasing this value you will get a graphic refresh of a large number of objects (for which the timer is managed), regardless of the amount of resources being used. It may be necessary to increase this value when the screen contains many objects animated with fast blinks which blink with synchronization.

### Slow Timer Tick Animation

This edit box is used for entering the frequency in milliseconds with which the screen's "average" and "slow" timer tick is to be double-checked. This value does not change the average or slow timer tick (default 1000 e 2000 milliseconds) but changes the frequency with which these animations are controlled. Therefore, by reducing this value you will get higher precision in the timer, regardless on the amount of resources being used.

### Slow Timer Tick Loop

This edit box is used for entering the value which expresses the maximum number of animations managed for each slow timer tick. Each tick is executed with the frequency set in the "Slow Timer Tick" property. By increasing this value you will get a graphic refresh of a large amount of numbers (for which the timer is managed) regardless of the amount of resources being used. It may be necessary to increase this value when the screen contains many objects animated with average and slow blinks which blink with synchronization.

## Screen Style Properties

---

By using this Style property you can set the Screen's parameters. In order to do this you only have to select the and open the required Screen and then edit its settings through the Movicon **'Properties Window'**.

### Keep Always in Memory

When activating this option, it will be impossible to close the Screen absolutely during project processing which can only be reduced to an icon instead and made invisible so as not to create any problems in displaying other open Screens on screen. As a consequence the window, even though not displayed, will continue occupying memory and system resources with the advantage of being of being rapidly displayed on screen when recalled. If this option is deactivated Movicon will destroy (unload from memory) the Screen window when not active and reduced to an icon (therefore becoming unusable) thus freeing memory space and improving resources in proportion to the complexity and size of the destroyed Screen. This will, however, cause the Screen window to take more time to display but nevertheless precious memory will be saved especially when many Screens are being used.



Screens checked with the **'Not Destroyable'** option will **NOT** be automatically loaded at project startup, but only when requested. After the first loading they will continue to remain active in memory.



It is advised to enable this **"Not Destroyable"** option only when effectively necessary. Usually this option is enabled for 'heavy' Screens which may cause a slow down in page loading in Runtime. Another reason for enabling this option would be when logic has been inserted into the Screen that must remain running even when the page is not displayed.

### Fit in Window

This property automatically adapts the Screen's size to fit into the window containing it, and therefore the screen resolution set in the graphic card. As a consequence the vectorial drawings and controls contained in the Screen will also be resized and therefore readapting the whole page according to the screen's resolution.

### Screen Colors Type

This property allows you to select a set of colours to be used for the screen. The colours set will be used by the Server project to display the page locally and to update the page on a Web Client, if any.

Possible options are:

- Use Device Default
- Black and White



The 3D-look buttons should not be used in a screen where the "Black and White" option is selected, since they will be displayed as completely black.

### Enable Scrollbars

When activating this option box, the system will allow the lateral scrollbars to be displayed when the drawing's size is bigger than the window's size. Otherwise the scrollbars will not be available in the window even when the drawing is bigger than the window.

### Show On MDI Tab

By activating this option box the Screen's MDI Tabs will be displayed during Runtime as well.

### Show Synapses

By activating this option box the Screen's Synapses connectors will be displayed during Runtime as well.

## Screen Background Properties

---

By using the Background properties you can define the properties concerning the Screen window's background. To do this just select and open the required Screen and then edit its settings through the Movicon **'Properties Window'**.

### Back Color

You may find it useful to set a different background colour to that proposed for default.

Screen backgrounds do not have to be made up of drawings: Movicon Screens can also be made up with controls only (buttons, displays, graphics, etc), therefore the background need simply be a uniform colour.



Background colors can be very useful when the Screen's drawing does not measure up to the required sizes for the Screen window. Therefore by using a background colour you can effectively hide the difference between the window's borders.

### Background Image

By using this dialog box you can associate a static image in BMP, JPG, GIF, WMF type, etc. as the Screen's background. As an alternative to entering the path and name, you can select the file by using the '...' browse command button found at the side of the edit box.



Pressing the "CTRL + (...)" keys consents to editing the selected background image, by opening the predefined application associated to that image in a new window.

### Image X pos

This edit box is used for entering the X coordinates in pixels of the eventual background image's position associated to the Screen window.

The coordinate refers to the position of the image's highest left corner in relation to the Screen's highest left corner (being X0).

**Image Y pos**

This edit box is used for entering the Y coordinated in pixels of the eventual background image's position associated to the Screen window.

The coordinate refers to the position of the image's highest left apex in relation to the Screen's highest left apex (being X0).

**Close Screen Delay**

Delay in milliseconds before Movicon unloads the Screen from memory after changing pages. When executing a change page with two Screens open in MDI mode, the one which is closed is only hidden and will be destroyed after this time has been exceeded (unloaded from RAM). When the value is set at '0' the Screen will be unloaded immediately and then the next screen will be loaded.

**Gradient**

By using this selection you can set the background colour shades associated to the Screen. The shades of colour can be selected from those proposed on the drop-down list (horizontal, vertical shades etc).

The shades of colour are a combination of the '**Background Color**' associated to the Screen and the one selected from the 'Gradient Color' property.

The '**none**' option means that no shades of colour have been associated to the Screen.

**Gradient Color**

By using this selection you can define the colour for the Screen's background gradient. The resulting various shades of colour will be a combination of the '**Background Color**' associated to the Screen and the one selected from the '**Gradient Color**' property.

**Image Width**

By using this edit box you can define the width in pixels with which the background associated to the Screen is to be displayed. When using the '-1' value the image will be displayed with its default size, otherwise the image will be resized according to the sizes set but with the possibility of losing its graphical quality.

**Image Height**

By using this edit box you can define the height in pixels with which the background associated to the Screen is to be displayed. When using the '-1' value the image will be displayed with its default size, otherwise the image will be resized according to the sizes set but with the possibility of losing its graphical quality.

**Tile Image**

If the associated file as background colour is the right size this selection permits the image to be repeatedly displayed along side each other in the Screen until the whole area in the window is covered.

**Spawn Thread Execution**

Each Screen processes the logics and tasks associated in the User Interface thread. When this option is active, the Screen will execute in a separate thread, independently from other project threads. This is handy to use when there are Screens containing significant logic or synapses processes. In this case the logic processing will be executed in a separate thread without penalizing the graphic interface, which nevertheless requires a greater commitment of the memory resources.

**Exclusive Write Access**

Enabling this property, only the first user who get connected to the screen resource can have write access to that screen. All other users who get connected can only open the screen but can not give commands, like setting a variable value using a display, potentiometer, etc. or acting on button commands like page change etc. This mode allows to avoid that simultaneous connections can write the same variable. As an example, if in Server project a user has opened the "LayOut" screen, any other WebClient user who get connected to the same screen could not have write access to that screen. Vice-versa the rule is valid even if the WebClient user first accesses the screen. In this case the local user of the Server project will not have write access to that screen. When a user opens a screen already opened with exclusive access by another user, a warning message will appear, as follows:

The screen 'ScreenName' cannot be opened with the exclusive Write Access

## Screen Execution Properties

---

By using the Execution properties you can define the Screen's advanced functionalities. In order to do this just select and open the Screen required and then edit its settings through the Movicon **'Properties Window'**.

### Layer Variable

If required by using this edit box you can select a variable from the Movicon Real Time DB which determines the number of display layers for the Screen's controls and symbols. If you want to manage the layers, each Movicon drawing or symbol must be declared with the required level by using the **'Layer'** property from the control's or symbol's **'Properties Window'**.

A Screen can have up to 32 layers. Therefore it is essential that the variable defining the number of layers must be in DoubleWord (32 bit), keeping in mind that each single bit corresponds to the displaying of the layer relating to that number, starting from zero bit which corresponds to the first layer.



Activating one layer will not exclude another, which means that there can be many levels active at the same time based on the bit set in the logic state '1' in the associated variable. In addition to this, the 16 available levels do not have any priority hierarchy over each other.

### Public Source Container

The base Screen's name for the Public Symbols is entered in this dialog box. The Movicon controls and symbols can in fact be associated a 'Public Name'. When some symbols with the same 'Public Name' have been inserted into the project, and one of these is present in the 'Public Symbols Container', all the public symbols will be modified with the base symbol's properties, being the one contained in the Screen specified in the 'Public Symbols Container' property, at the Runtime startup. These modifications only take place in Runtime, therefore the symbols will resume their original properties when returning back into planning mode.

### Style Source Container

By using this property you can select a screen to use as a style source container. For further information on how this property works please refer to "Style Sources in Symbols".

### Global Container

Enabling this property permits the Screen's child objects be published in DOM format (Document Object Model, XML).

### Style Source Back Color

By enabling this property, the screen will inherit the back color of the active style source container. For further information on how this property works see "Style Sources in Symbols".

### Execute Synapse

When enabled this property activates the execution of any synapses logic upon Screen loading.

### Synapses Cyclic Execution

When enabled this property activates the cyclic execution of any synapses logic when the Screen is active.

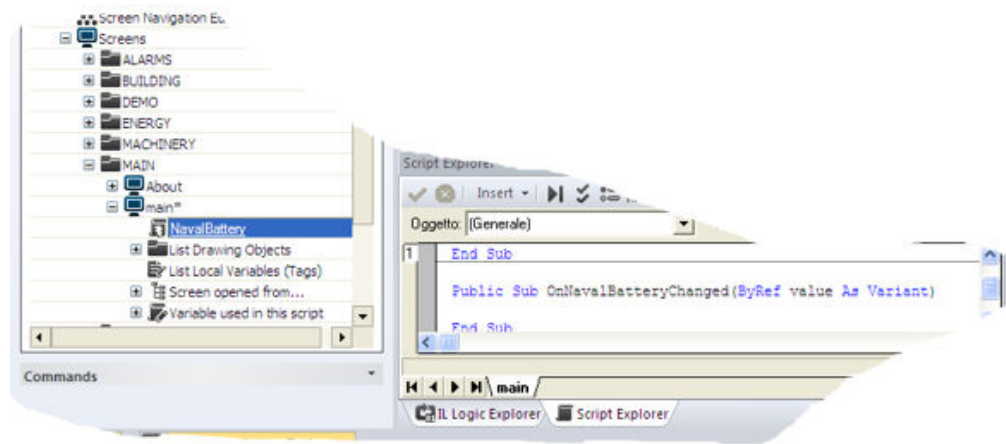
## Associating Variable Script Events to Screens

---

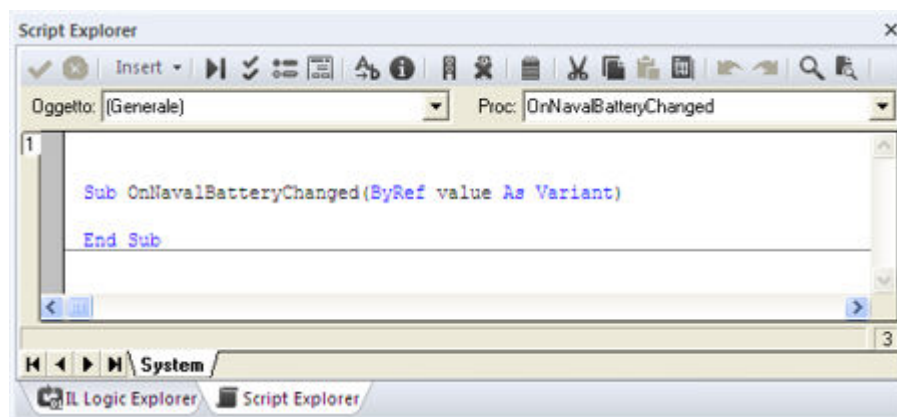
In addition to the standard events made available by Movicon (Click, DbClick, KeyDown, etc.) customized events, associated to Movicon Real Time DB variable variations, can also be inserted into the screen's Script Code. Practically, an event can be added to be executed whenever the selected variable changes status. Therefore it is the programmer's

task to insert the appropriate code into the event's internal according to what is required. By inserting the event in this way it will be active and processed only when the Screen is active, meaning when loaded in Ram.

Associating a Variable Script Event to a Screen can be done by selecting the Screen and activating the 'Add New Variable Script Event' which can also be found in the 'Command' window of the 'Project Explorer'.



The request for a Variable Script Event association to a Screen will open up a further window containing a list of the available Real Time DB variables. Once the variable has been selected the new event will automatically be inserted in the Script Explorer window with the syntax *"OnVariableNameChanged"*:



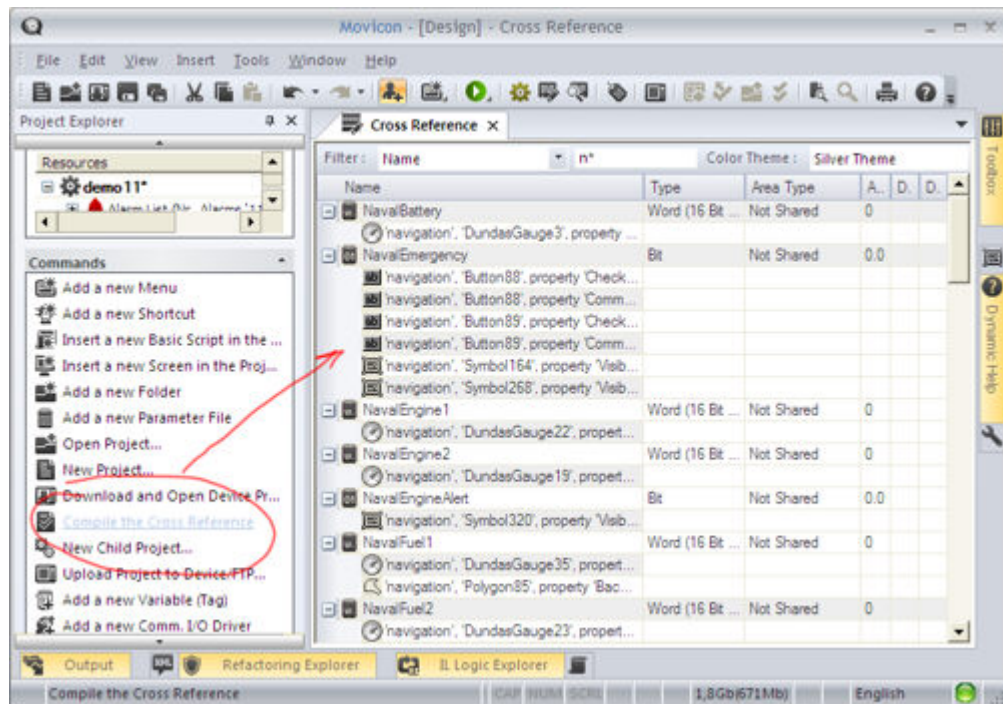
## 7. Cross Reference

*The Movicon Cross Reference list can be applied to both the project Variables and Screens and consents you to find out where a Variable is being used or from where the screen was opened.*

Movicon consents you to generate a Cross Reference List applied both to the project variables and screens. This functionality may be handy for quickly finding out in which resources or objects the variables and screens are being used and can be useful for printouts and documentation.

### Variable Cross Reference

The Cross Reference for variables allows you to single out the resources or objects which use specific variables. To compile or update the variable Cross Reference you can use the **"Compile Cross Reference"** command found in the Project Explorer's "Commands" pane or in the menu which opens by right clicking on the Real Time DB resource. Once the Cross Reference has been compiled a window will open within the workspace similar to the "List Variables Window", in extended form, but containing only those variables used with each one showing a link to the object or objects in which it is being used. You can also access this window by double clicking on the "List variables (Tags)" resource while keeping the SHIFT key pressed down. Another way in finding out where a variable is being used is by clicking the ""Variables used in ...." tree item available for each variable listed in the "List Variables (Tags)" resource.



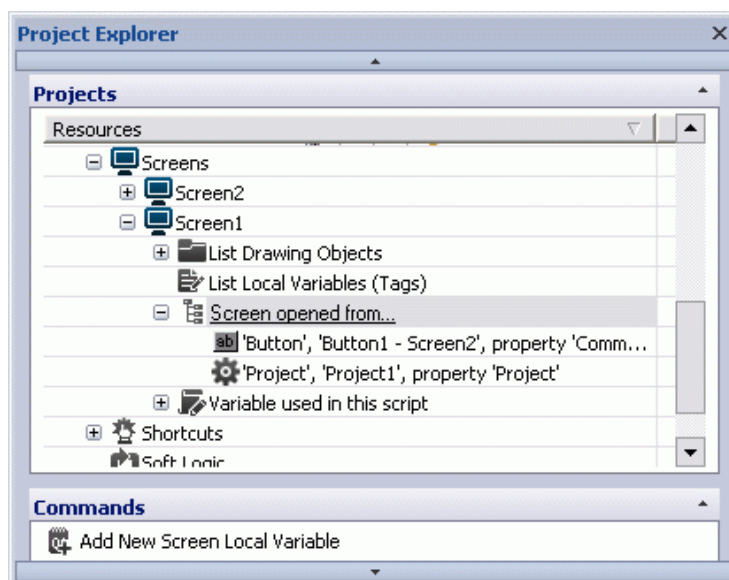
The variable Cross Reference has effect in all the project resources, even for those variables used in Communication Drivers, therefore in both Stations and Tasks. In addition to this you can also compile Cross References for variables used in Basic Script codes on condition that they are used directly without using the "GetVariableValue()" and "SetVariableValue()" functions.

Once the Cross Reference window has been opened showing a variable that has been used in an object contained in a screen, by double clicking on the link, this screen will open focusing on the object in question, and the "Dynamic Property Explorer" window will open on the property to which the variable has been associated.

## Screen Cross Reference

---

The Cross Reference for screens allows you to single out the resources or objects containing commands to open Screens. Unlike for Variables, a window does not display showing a Cross Reference for Screens. Instead, you can view the cross references of each Screen by expanding the "Screen Opened from..." tree item available for each Screen in the Project Explorer window. When this item is expanded (click on "+" node), and the Cross Reference has not yet been generated, you will be requested to compile and confirm the screen's cross reference which will then appear listed under the "Screen Open from..." item. You will then be able find out which resource or object contains a command to open the screen.



If the "Screen Opened from..." item expands (click on "+") already with a cross reference generated, just update it by using the same command used for the variables **"Compile Cross Reference"** which can be found either in the "Commands" pane at the bottom of the Project Explorer window, or in a menu which opens by right clicking on the Real Time DB Resource.



## 8. Graphic Interface

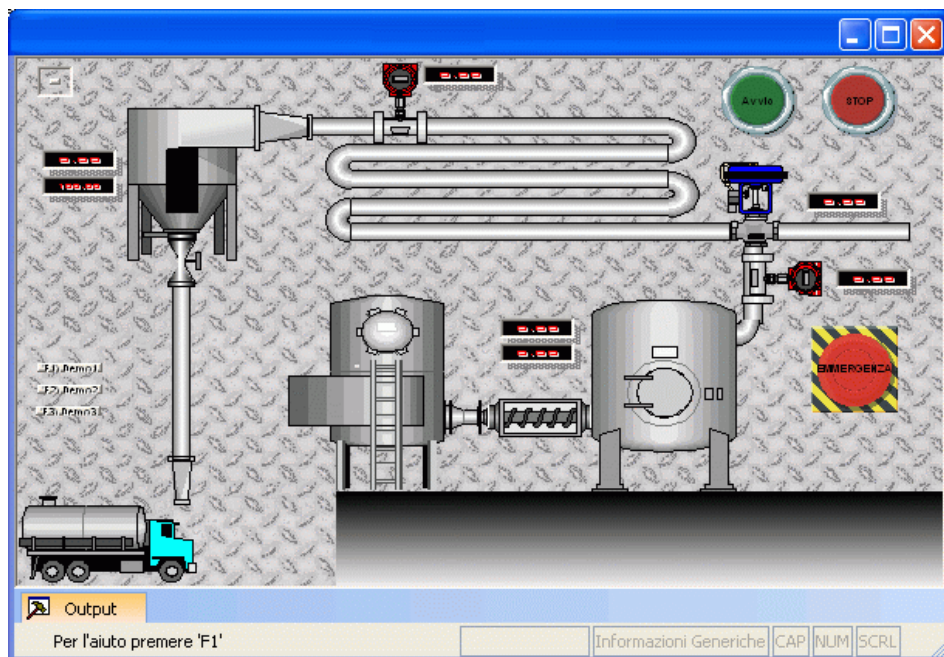
*This chapter introduces the different techniques that can be used for creating graphic interfaces and drawing in Screens*

The Movicon Screen resources are elements through which man-machine graphic interfaces can be created by using the drawing editor and graphic animation editing tools. Movicon permits drawings to be created in two different ways: by using its own internal graphic editor or by inserting drawings ((bmp, jpg, gif, wmf, emf, etc.). Both ways can be used when working with graphics as they can co-exist together.

The Movicon Objects and Controls which can be inserted on screen are available from the **"Objects Window"**. These components can carry out different functions, from simple geometric shapes to advanced controls for executing commands or displaying data.

The Movicon Objects and Controls are created in proprietary vectorial format and symbols can also be created (composed of a number of different components) and associated with animation properties. These formats can be exported or imported from Metafile (WMF, EMF) format.

Your own Movicon vectorial drawings can be saved in the **"Symbols Library"** and reused.



An example of a screen page using a background image and a number of vectorial symbols taken from the Symbol Library



# How to create Graphic Interfaces

---

Before going ahead with the animation procedures using the graphic objects and commands, it is always a good idea to establish which type of graphic solution you want to use, even though by choosing one solution does not mean the others cannot be used.



A 'classical' solution is usually to associate a static drawing file, such as a screen background, and then insert graphic objects and commands on the screen to be used for animating. These objects can naturally include the Movicon vectorial symbols.

The static drawing can be built from a file in BMP, JPG, GIF format etc. or from a vectorial file in WMF, EMF metafile format. The background drawing file can be graphically edited by using the appropriate image editor such as the Windows 'PaintBrush'. These programs can be linked to the Movicon **"Tools Menu"** so that they can be run straight away.

The background file constitutes the screen's static drawing, while animation is implemented by using the Movicon object techniques, which will be superimposed onto the background drawing.

It is not compulsory for a screen to be associated with a background file as a static drawing. The screen's graphic interface can also be created with just a drawing and the Movicon vectorial symbols.

The Movicon tools also permit you not to use any background drawings if you wish not to do so.

The drawing can be made up from lines, shapes or complex symbols which are already provided by Movicon, some of which can be coloured, customized or animated by using some of the symbols' or vectors' property windows.

## Editing technique Synthesis

---

The editing techniques use as base the insertion of object or symbols by means of purposely designed toolboxes or object libraries. Each single potentiality, function and property of each object are described in detail in properties window relating to each specific graphic object.

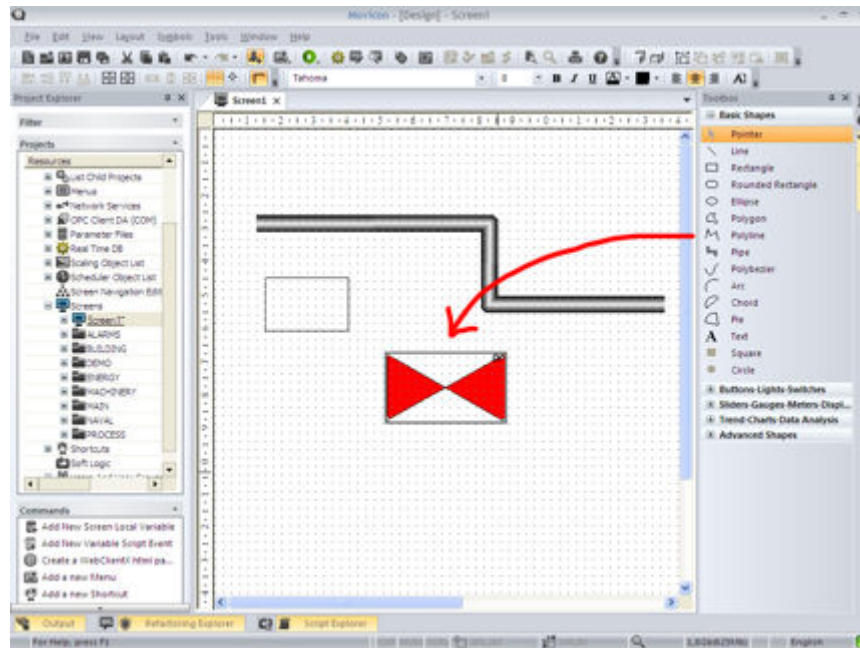
The synthesis of the graphics editing techniques are provides below and will then be described in detail for each single component.

The synthesis aim is to give you a general overall picture of the operations to be executed in created screens.

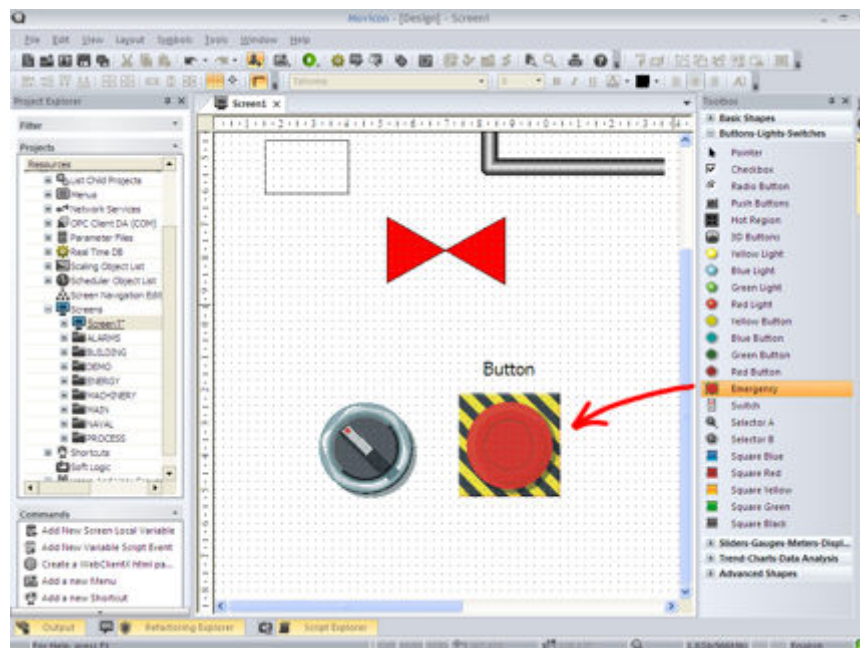
### Inserting objects into screens

This operations consists of taking out the desired object from the Toolbox and inserting it in the point desired within the screen area.

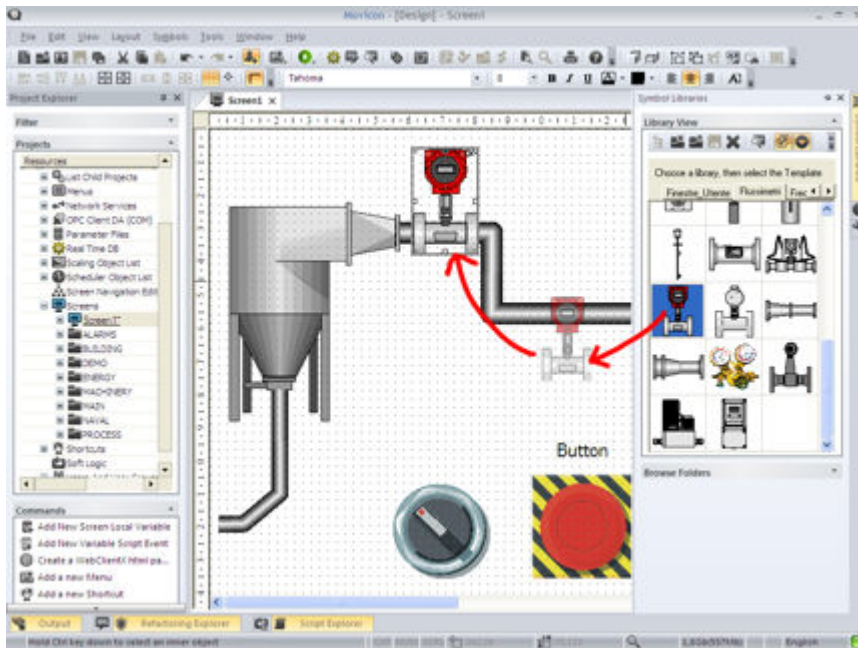
After having inserted the object, it can then be sized and configured as pleased by using the object's borders and properties.



The objects from the Toolbox include pre-built graphics such as buttons, selectors, sliders, meters, gauges, etc.  
In this case, according to the object inserted, you can set up a specific configuration by using the properties relating to each object.



Movicon also has Symbols Library containing countless ready-made graphic symbols. The graphic libraries can be displayed from the "View > Symbols" menu. The graphic libraries reduce time in creating any type of graphical project for automation. A vast choice of graphic symbols (tanks, pumps, valves, motors, etc) allow you to high graphical impact drawings with ease.  
Another thing to keep in mind is that you can import external drawings or you can expand the library even more by building your own custom graphic symbols.

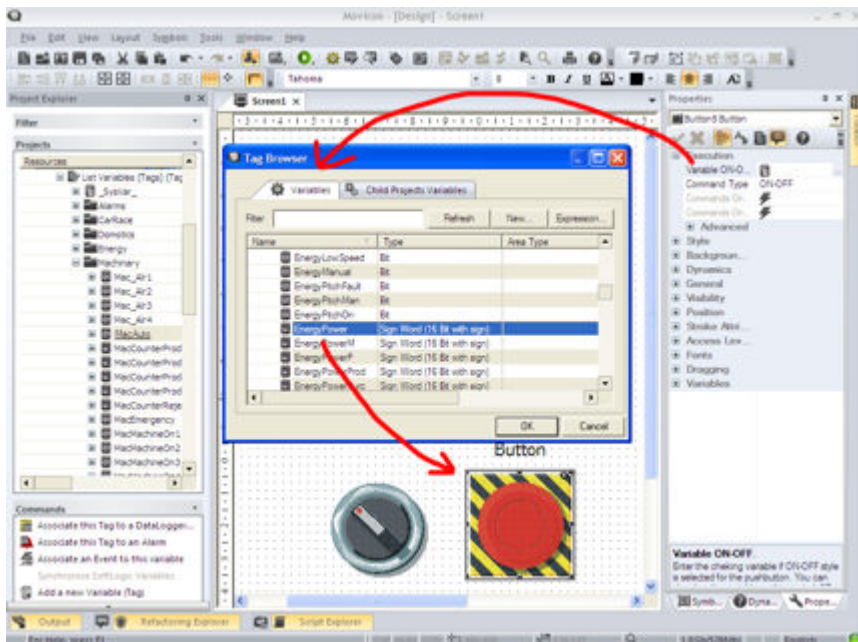


### Assigning variables to dynamic functions

After having inserted a symbol or an object, you can access its properties to configure it. Configuring is done through the Properties Window, which can also be activated by simply double-clicking on the object.

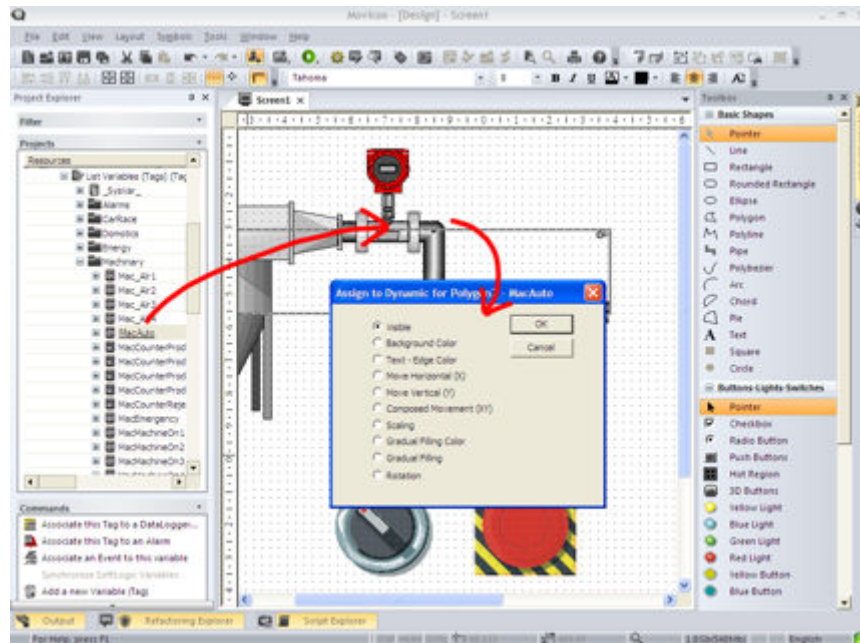
To make drawing or symbol more dynamic you need to use its 'Animation Properties', where you can assign variables for the graphical function desired along with other dynamic settings (ie. colors or color changes, thresholds for fillings or visibility, etc.).

The variables can be selected through the "Tag Browser" window which allows you to select (or to create directly) variables from the project's Variable List(Tags).



Assigning a variable (Tag) to a graphic object can be faster dragging the variable directly on the selected object, using the "Drag&Drop" method.

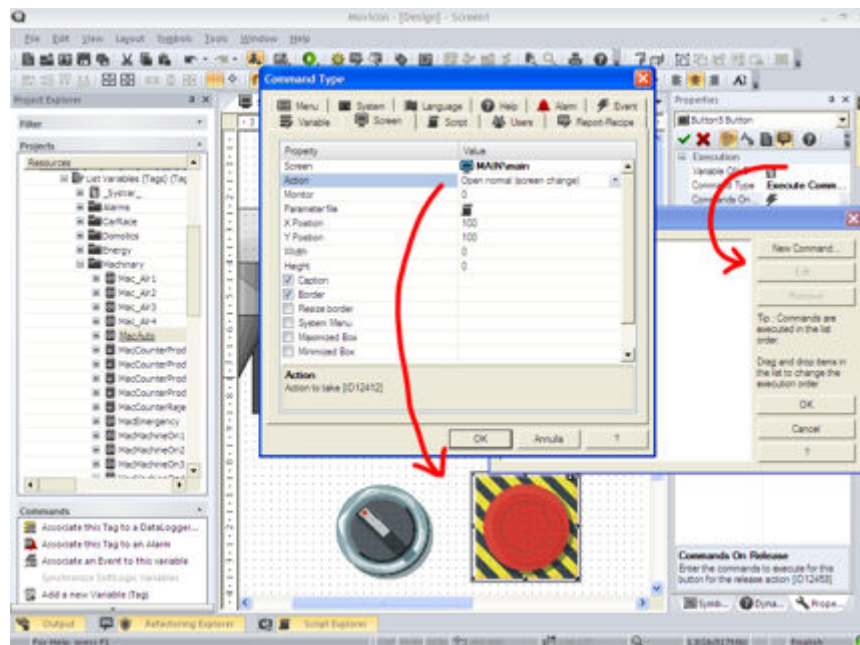
Just select the variable from the Project variables list and drag it to the graphic objects, as shown in the picture. If the object is a command-type object (i.e. a button, a potentiometer, etc.) the variable will be directly assigned to it. If the object is a graphic symbol, you will be asked to select one of its dynamic functions. You can specify more details (colours, thresholds, etc.) later, acting on the object properties.



### Assigning commands

The command type graphic objects (such as buttons, but in the same way for menus, accelerators, events...) can be associated with one or more commands to be executed by clicking on the object.

In this case the command type window used for assigning commands can be displayed by **SHIFT+double clicking** on the object, or by using the 'execution properties'.



For any further information on the editing techniques and the object properties, please refer to the description on each object in this manual.



# Graphic Editing Operations

---

Any drawing object inserted into the screen can be edited and handled according to the options provided by the graphic editor. The drawing is entered into the screen after being taken out from the library then by clicking the initial insertion point and releasing the mouse key when having dragged it to the size desired.

The inserted object can be repositioned by simply selecting and dragging it with the mouse to the new position.

To resize the object just click on one of the sizing points and drag it until the desired size is reached. To introduce or modify a text directly in the drawing, just select the drawing and click on it. The cursor, indicating the possibility to type the desired text, will appear.

When the text has been entered press ENTER or press ESC key to cancel operation. The text entered will become the object's title and can be verified and modified through the **"Properties Window"**.

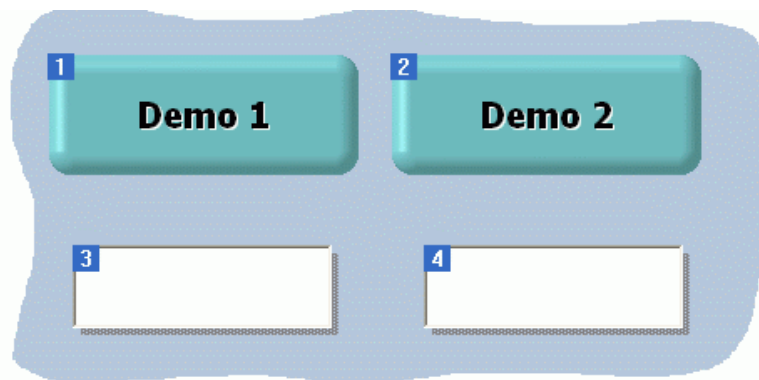
## Tab Order

---

The order of insertion always determines the Tab order of the objects inserted in a screen window. The Tab Order determines the sequence of selecting objects by using the TAB keys on the keyboard.

The TAB order can always be changed by using the **"Tabulation Order... (CTRL+D)"** command from the Layout Menu or the CTRL+D keys on the keyboard.

The displayed order can be changed by clicking the mouse on the object repeatedly to get the order desired.



The object with the highest Tab order will also be the one with the highest overlapping order. placed on top (at the forefront) of the thelayer highest.

## Overlap Order

---

The objects inserted on screen can be overlapped differently to the one they were inserted with.

To change the overlap order between the drawing object you need to use the commands from the **"Symbols Menu"**. These commands are used to change the objects' tabulation numbers. The commands available are:

- **First:** the selected object will be placed on the top layer in respect to the others (foreground). Therefore its tab number will be the highest number
- **Last:** the selected object will be placed at the back on the last layer (background) in respect to all the others. Its tab number will therefore be the lowest number

- **Move Next:** the selected object will be placed on top of the last layer. Its tab number will increased by one. The previous object will acquire a lower layer to which it had before
- **Move Previous:** the selected object is placed on the layer underneath the top one. Its tab number will be decreased by one. The previous object will acquire a higher layer to which is had before

## Multi-Object Selections

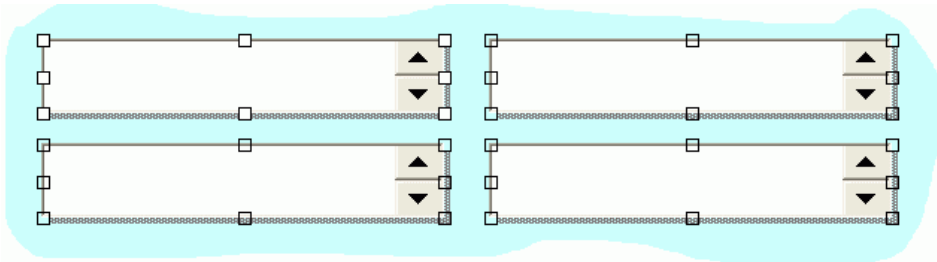
When using elements within a synoptic, it may sometimes be convenient to select an area containing a group of elements to modify their positions, alignments or sizes or select the Cut or Copy functions.

To select a group of elements in a synoptic, use the Windows standard techniques:

1. Click with the mouse on the initial point of the area you wish to select. Hold down the left button and move the mouse. A dashed extendible outline will appear to define the selected area. Move the mouse until the required dimension is reached
2. The Symbols can be selected also by clicking with the mouse on each individual symbol while holding down the CTRL key. The first symbol selected will be taken as reference

The commands you can use after having done a multi object selection are:

- Click on an object from the multi-selection and set it as the reference object
- Ctrl+Click on an object from the multi-selection to unselect that object
- Ctrl+Click on an object not included in the multi-selection to add it to the multi-selection



The reference control in a symbol group is the one that shows the delimitation border not transparent, but with solid black back color.

## Object Alignment Functions

The drawing objects inserted on screen can be can be exposed to the alignment and sizing functions which refer to one object or a group of objects.



The alignment functions are available when selecting in sequence two or more objects on screen by clicking the mouse with the CTRL key pressed down. The first selected object will become the reference object, which all the alignment or sizing commands will refer to. Once the group has been selected, the reference object can be changed by clicking it.







**The Alignment functions also allow selected objects to be aligned according to their barycentres. This can be done by pressing the "Shift" key and activating the command from the "Aligning Bar" or "Layout Menu". The command always considers one central barycentre for the selected objects.**

The commands which can be used are available from the **"Layout Menu"** and in the **"Aligning Bar"**.

## Alignment Objects

---




	Left	This command aligns the selected objects on the screen's left hand side.
	Right	This command aligns the selected objects on the screen's right hand side.
	Up	This command aligns the selected objects at the top of the screen.
	Down	This command aligns the selected objects at the bottom of the screen.
	Centre Vert.	This command aligns the selected objects in the centre of the screen vertically.
	Centre Horiz.	This command aligns the selected objects in the centre screen horizontally.



**The object with the lowest Tab number will be used as the reference object for aligning the other objects. The reference object's small selection squares will be highlighted in a different colour in respect to the other objects. Once the group of objects has been selected you can change the reference object by clicking the one you desire.**

## Set Same

---



	Width	This command resizes the selected objects to the same width on screen.
	Height	This command resizes the selected objects to the same height on screen.
	Both	This command resizes the select objects with the same width and height on screen.



**The object with the lowest Tab number will be used as the reference object for aligning the other objects. The reference object's small selection squares will be highlighted in a different colour in respect to the other objects. Once the group of objects has been selected you can change the reference object by clicking the one you desire.**

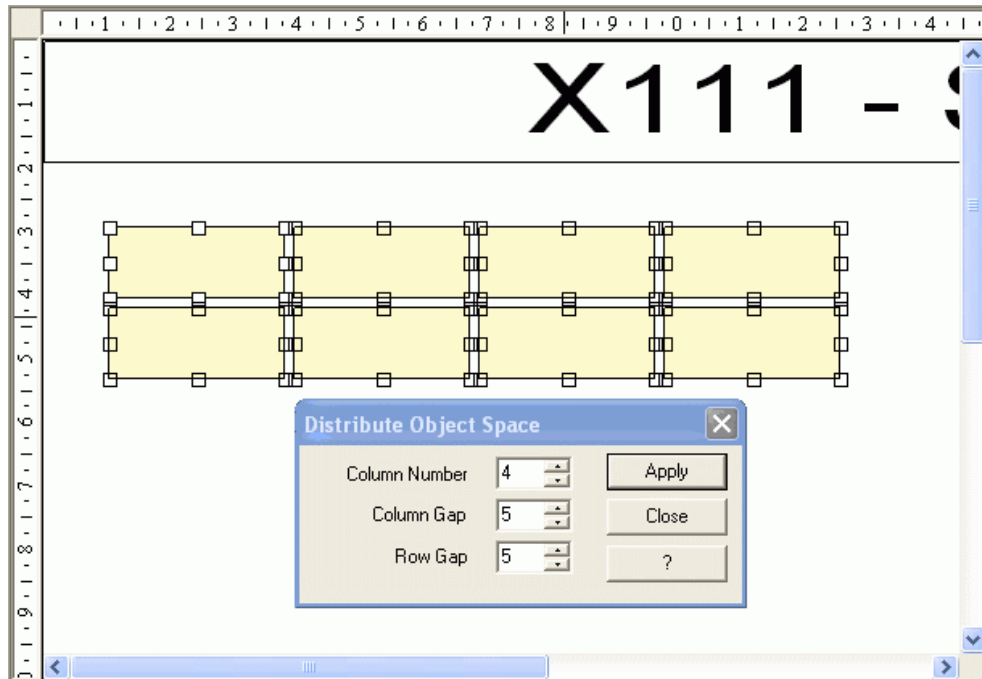
## Center in Window

---

	Vertical	This command centers the selected objects vertically in the screen area. When more than one object has been selected the area occupied by the selected objects will be centered.
	Horizontal	This command centers the selected objects horizontally in the screen area. When more than one object has been selected the area occupied by the selected objects will be centered.

## Distribute Object Space

The **"Distribute space between Objects..."** function, from the Edit Menu, is one of the most important tools. This graphic editing function aligns screen objects equally. In this way the symbols will be positioned automatically by being subdivided and organized in rows and columns equally with the possibility to preview the result.



The 'Distribute Object Space' automatically aligns the symbols in rows and columns

To get this function working you must first select the group of objects you wish to align with the usual standard techniques. After having done this you can then access the **"Distribute space between Objects..."** command from the Edit Menu. A window will display for setting the following parameters:

- **Column Number:** sets the number of columns in which the group of symbols are to be subdivided
- **Column Gap:** sets the number of space pixels between the symbols and the set columns
- **Row Gap:** sets the number of space pixels between the symbols for the set rows

You can get a preview of how these settings will look like, by using the Apply button which will distribute the objects immediately.

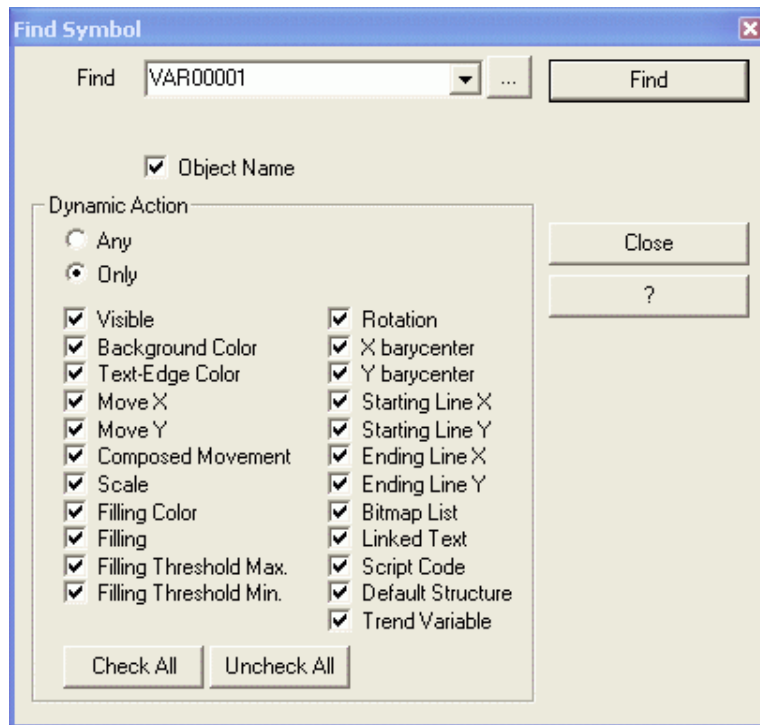


## Find and Replace symbol

When a screen is opened in edit mode in the workspace, the "Edit" menu includes two commands, "Find Symbol..." and "Replace Symbol...". They can be used to find if a variable is associated to one or more screen objects and, if required, to replace the variable in the objects with another one.

### Find Symbol...

This command will open the following dialog window:



The command permits to establish if a variable has been used in animation properties for objects added to the screen currently open in edit mode. The fields in the above windows have the following meaning:

**"Find"**: type in this field the name of the variable to seek in Animation properties or in objects' Basic Script code. The drop down list includes the texts already searched before. Clicking on the ellipse button "..." on the right of the drop down list, the Tags Browser dialog is open, showing the project's variables list.

**Object name**: if this check-box is selected, the text typed in the "Find" field will also be searched in the "Object Name" screen objects property. This way you can search an object name instead of the variables used in it (in this case the text in the "Find" field can be not a variable name).

**Dynamic Action**: in this options group you should define which object properties will be searched in. You can select all Dynamic Action properties or just some, you can also extend the search to the objects' Basic Script code selecting the "Script Code" option.

**"Find" button**: this command starts the search. When an object containing the search text is found, the search is stopped and the object is selected. Clicking the "Find" button again the search will restart until the next object (if any) is found.



The variables search in the Dynamic Animation properties is executed only among the enabled Animation properties. As an example, if a variable has been associated to the "Variable" field of the "Dynamic-Visible" property of an object, but the "Enable Visibility" property is not set, the "Find" command will not include in the search the "Dynamic-Visible" property of that object.



When the search is executed in the Basic Script code, the text typed in the "Find " field is searched in the whole script. This means that any word or part of a word containing the text typed in the "Find " field will be found, being a variable name or not.

#### Replace Symbol...

This command will open the following dialog window:

The "Find Symbol" dialog window is shown. It has a title bar with a close button. The "Find" field contains "VAR00001" and the "Replace" field contains "VAR00002". Both fields have a dropdown arrow and an ellipsis button. The "Object Name" checkbox is checked. The "Dynamic Action" section has two radio buttons: "Any" and "Only", with "Only" selected. Below this is a list of 20 checkboxes, all of which are checked. At the bottom of this list are "Check All" and "Uncheck All" buttons. On the right side of the dialog are four buttons: "Find", "Replace", "Replace all", and "Close". At the very bottom right is a button with a question mark.

Property	Checked
Visible	Yes
Background Color	Yes
Text-Edge Color	Yes
Move X	Yes
Move Y	Yes
Composed Movement	Yes
Scale	Yes
Filling Color	Yes
Filling	Yes
Filling Threshold Max.	Yes
Filling Threshold Min.	Yes
Rotation	Yes
X barycenter	Yes
Y barycenter	Yes
Starting Line X	Yes
Starting Line Y	Yes
Ending Line X	Yes
Ending Line Y	Yes
Bitmap List	Yes
Linked Text	Yes
Script Code	Yes
Default Structure	Yes
Trend Variable	Yes

The command permits to establish if a variable has been used in animation properties for objects added to the screen currently open in edit mode and to replace it with a different variable. The dialog window include the fields already described for the "Find" command, moreover some additional fields are present:

**"Replace":** type here the name of the variable to be used for replacement. The drop down list includes the texts already searched before. Clicking on the ellipse button "..." on the right of the drop down list, the Tags Browser dialog is open, showing the project's variables list.

**"Replace" button:** this command executes the replacement of the searched variable with the new variable specified in "Replace" field. You should first execute the "Find" command, and, as soon as an object containing the searched text is found, the search is stopped and the object is selected. At this point, pressing the "Replace" button, the found variable is replaced with the new variable, for the selected object only

**"Replace all" button:** this command executes the replacement of the variable specified in the "Find" field with the variable specified in "Replace" field. In this case you should not execute any search since the replacement is executed in all screen objects.

## Importing/Exporting Vectorial drawings

---

The Movicon graphic editor has to be proprietary type due to its intrinsic graphic animating objects characteristics.

However, in upholding the system's philosophy of openness, you are guaranteed the possibility of importing or exporting vectorial graphic images with the Movicon graphic editor.

For instance, you can import drawings in Windows' metafile format (WMF, EMF) created by any vectorial graphic editor. Once imported into a Movicon screen these drawings can be edited and configured in their graphic and animation properties offered by the system.

It is likewise possible to export Movicon drawing objects in one of the above listed formats, thus making them available to other vectorial graphic editors.

To export a Movicon drawing object you need to select it, then access the **"Export Drawings..."** command from the Edit Menu or Drawing Menu. The path and name desired then need to be specified in the standard window for saving files.

To import a WMF or EMF drawing on screen, you need to use the **"Import Drawing..."** command from the Edit Menu or the Drawing Menu. By using the appropriate window you then need to select the file name desired. The file will be converted and made available on screen as a Movicon object. The Zoom Factor will be requested before importing on screen, i.e. the factor for resizing the original drawing. If you wish to keep the original sizes confirm with OK or otherwise specify the conversion factor desired.

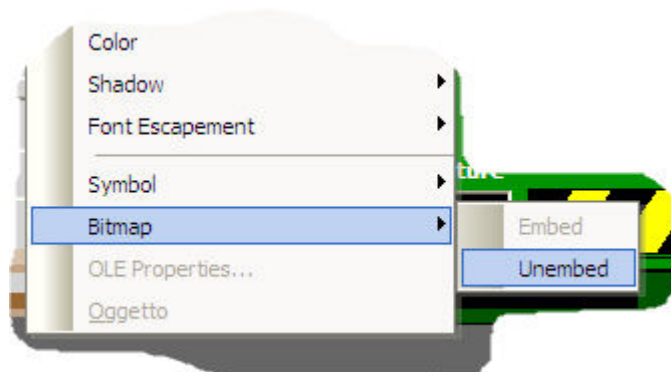


**When importing "wmf" or "emf" files, Movicon only supports the primitive Windows drawing functions. It does not support the Raster function which use the graphics editor. These functions are needed, in certain cases, for generating bitmaps in the drawing export phase, so when this is the case Movicon will not import these bitmaps. Therefore, it is therefore necessary that the file, to be imported, contains only primitive vectorial elements, such as lines, curves, etc.**

## Embedding Images in Symbols

---

This symbol function allows images to be embedded in symbols which have been linked to the symbol through the 'Fill Attributes' Group's "Image" property. The image or images will then be saved together with the symbol and the use of the image file will no longer be needed. The command for embedding the image can be found on the menu which appears by clicking the symbol with the mouse's right button:



If you execute the "Unembed" command the image file will be recreated by Movicon in the same path and with the same name it was embedded with. However, the original file path must still be in existence otherwise Movicon will create an error message.

## Locking-Unlocking Objects

The Lock function is a very handy thing for the programmer to have.

By using the Lock command, which is activated with the right mouse key, on the point where the object drawing is on screen will block any changes made with the mouse but not those done with the keyboard's direction arrows.

This utility can be exploited to avoid any accidental movements being made to the object on screen.

When the mouse pointer enters in the proximity of a locked symbol or object, a cursor in the form of a lock will appear along side it to show they have been locked.



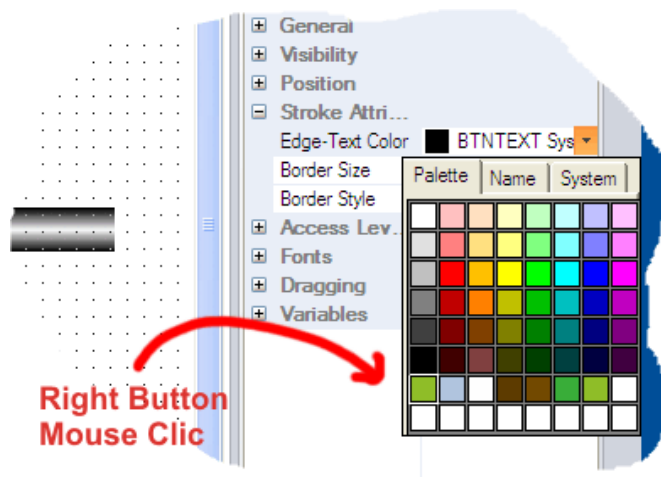
Re-click with the right mouse key on the lock command to unlock the object.

## Colours Selection

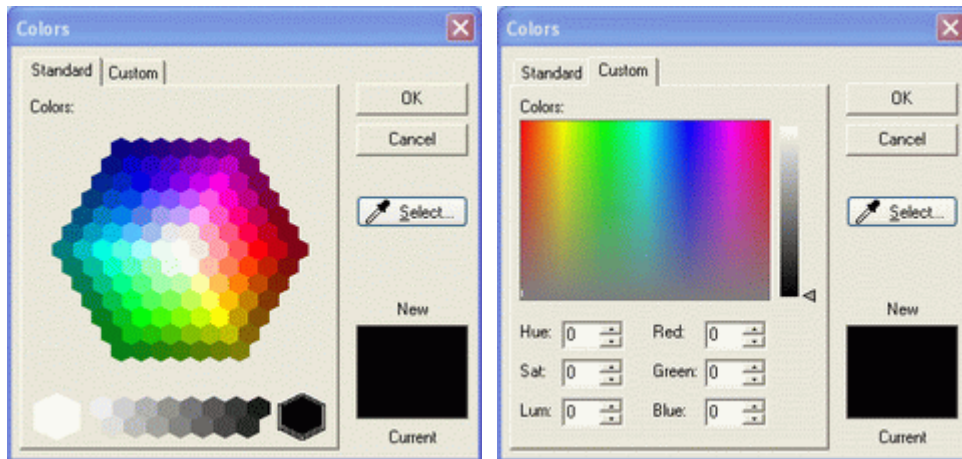
It is a general rule to set or select colours during the development of a Movicon project. The colours can be set through the 'Properties Window' of all those objects and resources which are provided with this possibility. The color settings are activated by using the purpose built button from the 'Properties Window' (the button showing a down arrow, placed on the right side of the property field) of the object or resource.

Movicon, in this case, will automatically display a popup window split in three tabs:

1. "Palette" tab. It shows the standard colours palette, where the 48 basic colours are represented. You can define 16 persistent custom colours right clicking on one of the squares of the last two rows, by default representing 16 white squares. Placing the mouse cursor on a palette square, a tooltip will appear showing the RGB colour code (being R, G and B decimal numbers ranging from 0 to 255).
2. "Name" tab. It shows a colours list specifying each colour name.
3. "System" tab. It shows the system's colours list.



When defining a new custom colour, right clicking on one of the squares of the last two rows of the "Palette" tab, a dialog window is open, showing two tabs. The first tab allows to select a colour among the standard colours, the second tab permits a complete colour customization, even typing the RGB colour code directly, as shown in the image below.



By using the "**Select...**" button in the window you can select a colour from the screen. When the button is pressed, the mouse pointer will change to a 'dropper' which draws up the color it is clicked on.

When the colour has been selected, it is saved in the object properties, using the colour name (if it is a standard colour) and the RGB code (3 pairs of hexadecimal numbers rrggbb, where rr=red component , gg=green component, bb=blue component). As an example, selecting a standard yellow, the object property will show a yellow square and the "Yellow (ffff00)" string .

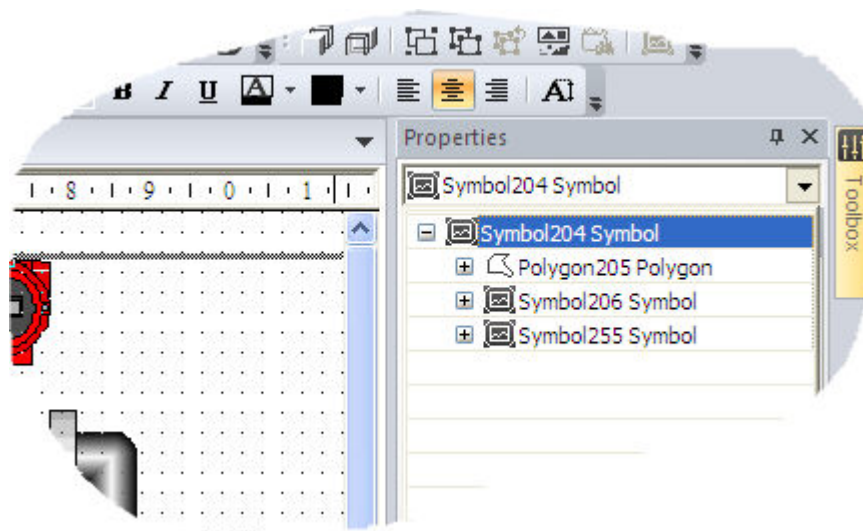
# Editing Symbol Graphics

The Movicon vectorial drawings allow you to enter drawing objects which can be treated as static objects, dynamic objects or vectors can be grouped together and treated as one single object. These sets of vectors, grouped together in one single object, are called **Symbols**.

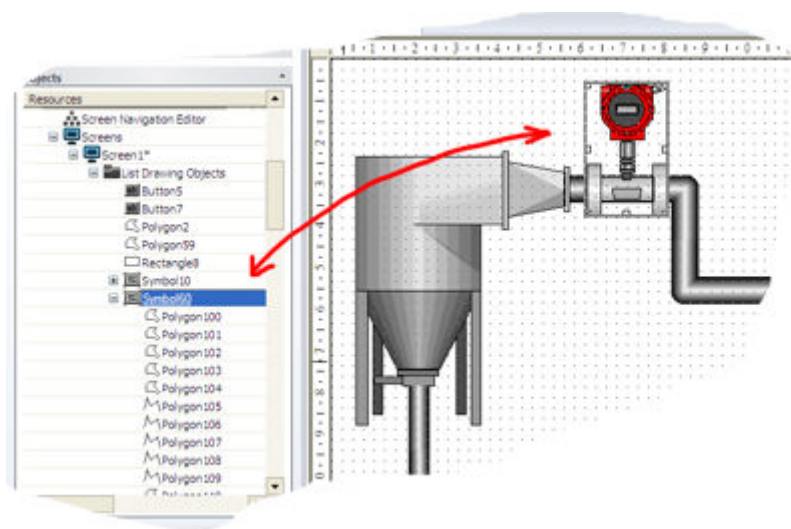
The Movicon Symbols can be edited and saved freely in an appropriate vectorial "**Symbols Library**", which will be linked to the installed system in the folder:

"..\Documents and Settings\All Users\Application Data\....\symbols"

The Movicon vectorial symbols have \*.msxz extensions.



Movicon has a vast library of pre-built vectorial symbols, subdivided into categories, to which the programmer can use to realize his/her own graphical interface in the screens. These libraries can be enlarged with new modified and customized symbols.



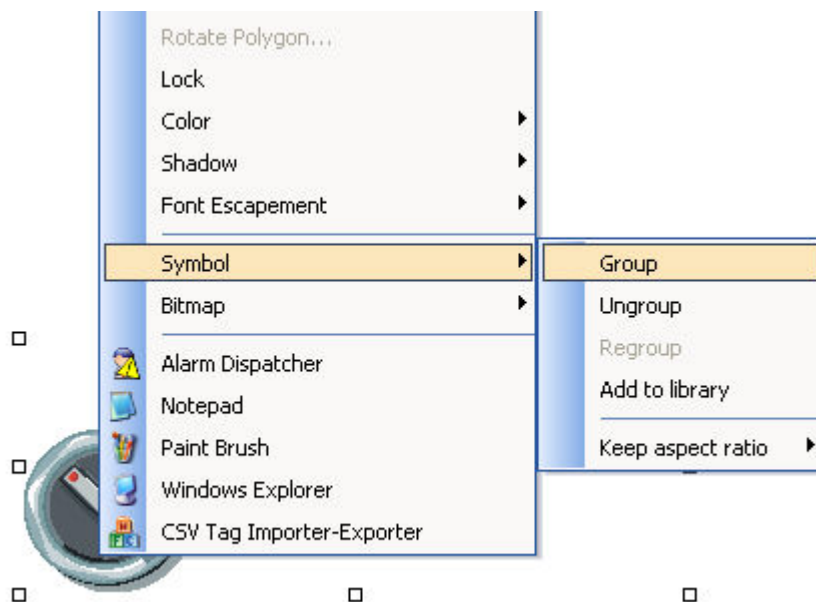
The images show a selection of a group of drawing elements. A group of drawings or symbols can be selected to create a new symbol, using the "Create Symbol"

command available right clicking on it. This way the drawing objects group becomes a symbol which can be saved into the library. The symbol structure can be seen and selected from both the combo box in the symbol properties window (see first picture above) and the synoptic resource in the "Project Explorer" window (see second picture above). The symbol structure components list allows to select and associate the graphic commands to each single element.

The command for graphically editing symbols can be accessed from the **"Symbols Menu"** or by right mouse clicking on the Symbol item.

## Commands for Creating Symbols

Creating Movicon Symbols is done by using the commands available from the **"Symbols Menu"** or by using right mouse key:



### Group

This command is made available when a vector or a group of vectors (drawing objects) is selected. To select a group of vectors you need to click on a point outside the group and drag the selection until all the desired vectors are enclosed. As an alternative you can click all the vectors singularly while keeping the CTRL key pressed down. This command is activated after having selected the group of vectors to create one object thus a Symbol. This symbol can then be added to the Symbols library as described below.

### Ungroup

This command is made available when a symbol is selected. This command, when executed, ungroups the symbol back to the original group of vectors which form it. The vectors are restored with their original sizes.

### Regroup

This command allows you to put the previously ungrouped symbol back together, without altering any links or names assigned beforehand. This function works when the symbol is a Power Template and uses the Basic Script functionality for linking the symbol's objects. The regrouping restores these links without altering the functions of the symbol before it was ungrouped. Symbol regrouping is supported up to the first grouping of symbols, any symbols added after and then ungrouped will not be regrouped together at the same time with the first grouped symbols, but after by using the regroup command again.



#### Add to library

This command allows you to enter the selected symbol into the **"Symbols Library"**. A selection window will appear when activating this command to choose the desired category.

The commands relating to the **"Symbols Library"** are described in the appropriate sections.

#### Keep aspect ratio

This command allows you to restore the symbol to its original size. This command works when the symbol has been resized differently from its original sizes.

- **Vertical:** sets the selected symbol at a vertical size in direct ratio to the horizontal size.
- **Horizontal:** sets the symbol at a horizontal size in direct ratio to the vertical size.

## Templates: Dynamic Symbols

---

Movicon allows you to use a technology which has been designed to simplify and decrease the time needed to building applications: the Power Template© technology. This philosophy allows the programmer to independently create his own objects, configure basic alarms, "type" recipes, Trends and Data Loggers.

After being configured as required, each symbol group can be saved in the Symbol Library by applying the "Template" concept. Each Template symbol can be inserted in an appropriate symbol category or in a new category created by the user.

This concept is the basis of the modern programming techniques to re-use and exploit previously created symbols in order to avoid having to go through the whole process again to create another one.

- **The Power Template technology allows Movicon to retain settings, animations, functionalities and any basic scripts associated to the symbols.**

When symbols are inserted on screen or in the project, Movicon will propose the automatic insertion of the variables associated to them with a default name and address. Therefore, those functions needed can be maintained and updated where necessary.

This is intended to save you a great deal of time in having to access and configure all the resources involved which can now be updated and managed completely in automatic.



Each single property set for each symbol or object will become part of the Template and saved in the symbol or object in the library. The symbols will maintain each animation feature or each VBA script and can be saved and used as a Template.

## Creating or Editing Symbols

---

New symbols can be created and added to libraries or already existing symbols in the library can be edited and customized.

In order to create a new symbol, you need to proceed as follows:

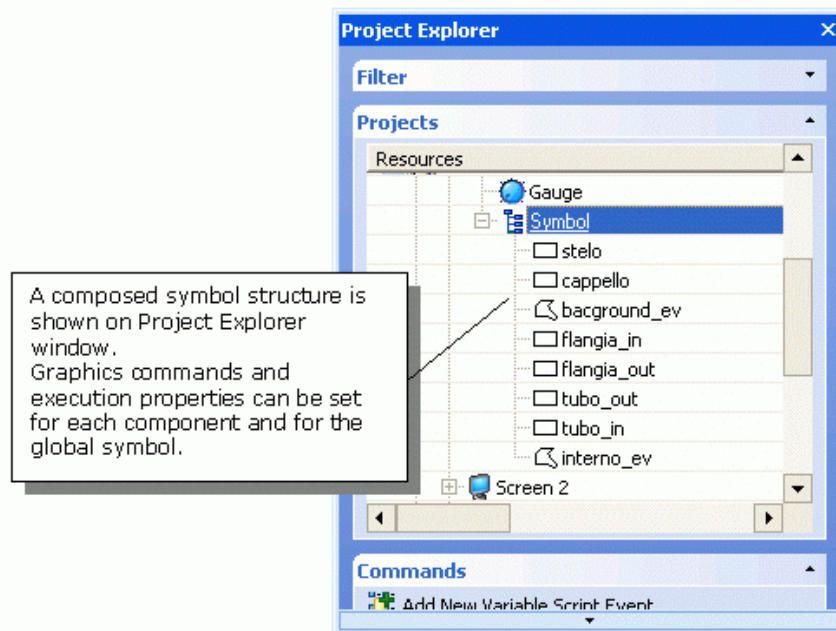
1. Design the symbol by inserting the various objects used for building vector components, until you get the graphical effect desired
2. Select the whole group of vectors by clicking on a point outside of the symbol and drag the selection to include all the vectors
3. At this point, activate the **'Symbol - Group'** command from the **'Symbols menu'** or activate the same command with the right mouse key



The symbol has now been created and is ready to be inserted into the library by using the appropriate command.

An existing symbol can be edited after it has been entered on screen, by selecting it and then activating the '**Symbol - Ungroup**' command from the '**Symbols Menu**'.

This will return the symbol back to its original form with its vectors ungrouped where each one can be edited as pleased.



A symbol, being a collection of drawing objects, is identified in a tree structure which shows a list of the parts it is built with.

The tree structure of each symbol can be displayed in the '**Project Explorer**' window.

The example above illustrates a symbol structure.

When double-clicking on one of the symbol's components in the 'Project Explorer' window, the symbol will highlight the corresponding component on screen so that it can be identified straight away.

The symbols can also be modified without having to decompose the symbol if involving the properties of already existing components. A composed symbol can be modified through its components by simply selecting the drawing which must be edited from the '**Project Explorer**' window. By doing this the Movicon '**Properties Window**' will be updated with the property of each component, which then can be edited. It is also to select a symbol's component by keeping the "CTRL" key pressed down and left mouse key clicking on the component. In this case the component will turn completely black and the Properties Window will show all its properties.

If the symbol is to be changed graphically, for instance drawings must be added or taken away, the only solution is to decompose the symbol and then recreate it.

## Resizing Symbols

Grouped symbols can be resized according to their original proportions. By clicking on one of the symbol's one-way directional sizing dots while keeping the "shift" key pressed down, the group will resize but will retain their original proportions.

## Inserting Symbols into the Library

You can add new symbols to the Movicon vectorial "**Symbols Library**", to meet any kind of customizing requirement you may have. Before inserting the symbol into the library you have to first create it by using the various vectors (lines or geometric shapes created with the Movicon graphic editing toolbox) which used together will compose the drawing of the symbol:

1. After having created the symbol, select the set of vectors with which it has been composed with and active the **'Symbol - Group'** from the **Symbols Menu** or by using the right mouse key. This operation will combine the set of vectors into one single vectorial. Use the **'Symbol - Ungroup'** to undo this operation.
2. After having created the symbol, activate the **'Symbol - Add to Library'** command from the Drawing menu or use the right mouse key to add it to the vectorial symbol library. By doing this a window will appear for selecting the category to which the symbol will be associated.
3. Select the symbol category desired, then insert the symbol by activating the **'Insert Template'** button. If you wish to create a new Symbol category, activate the **'New Library'** command in the **"Symbols Library"** window.



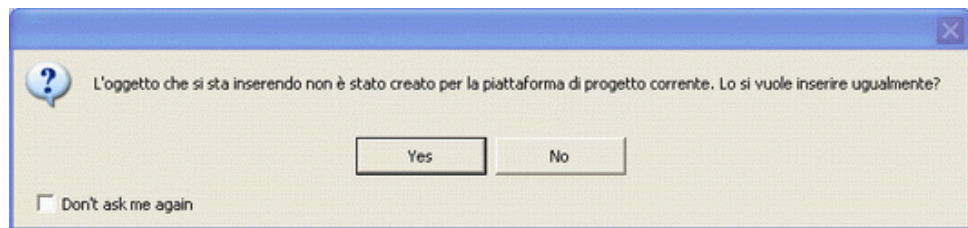
**The Symbol Library can also host symbols containing Active X object inside.**

## Inserting Symbols on Screen

To insert a graphic symbol in the screen you need to activate **"Symbols Library"** with the 'Symbol Libraries' command from the View Menu.

Next select desired category and select the template among those presented then press the 'Insert Template' button found on the window's right border. You can also drag your chosen symbol from the library directly to the point desired on the screen (Drag & Drop). After having insert the symbol on screen you can then size it as pleased by dragging its borders or position it to the point desired.

The following message may appear when inserting a Template:



This message means that the selected Template is not supported on the WinCE platform. This control is carried out by Movicon when in the following conditions:

1. The "Visible on CE Platform" option has been disabled in the Symbol's "Visibility" property group.
2. The "Windows CE" platform has been selected in the project's "Platform" properties.

When these two conditions are eminent, it means that the template is not enabled to function on WinCE and that the project where the template is being edited is destined for the WinCE platform. Therefore, it would not make sense to insert the Template in that project.

## Script Code Editing

The **"Dynamic Property Inspector"** window is a very powerful tool used for quickly editing Template script codes and is accessed by clicking the right mouse key on the Template.

This command opens a dialog window where a series of Tabs are presented at the top. Each one of these represents each single component composing the symbol, while the variables and script codes associated to the selected component are displayed in the center of the window.

This tool can be used for keeping an eye on and quickly editing the variables used in the drawing and also the script codes contained within them.  
The symbol's component's code can also be edited by selecting it from the **"Project Explorer"** window and then opening the **"Script Explorer"** window.

## Variables inside Symbols

---

When a symbol is exported to the **"Symbols Library"**, Movicon will also automatically export the variables which have been associated to the symbol. In this case Movicon will memorize the variable name and its data format. When the Movicon Real Time DB variables have been used in the symbol's or one of the component's basic script codes, the name of these variables must be written between two '#' chars. This is because Movicon has been able to distinguish which are the Movicon variables and which are the local variables within the basic.

Eg: `GetVariableValue("#namevariable#")`

This particular syntax is only done when the symbol has to be exported to the Library. A scrip code with the syntax, as shown above, cannot be executed and generates an error.



The Real Time DB variables in basic codes are not directly made visible by Movicon during the exporting of Symbols into the Library phase. They must be enclosed between two '#' characters before exporting the Symbol.

**When a template is inserted on screen, its variables will be created as local screen variables by keeping the SHIFT key pressed down. If the SHIFT key is not pressed down the variables will be created as global variables in the Real Time DB.**

## Images inside Symbols

---

When the symbol you want to export to the **"Symbols Library"** contains an image which must be exported together with the symbol, you first need to Embed the image inside the symbol before carrying out the exporting procedures. To do this simply click with the mouse right key on the symbol and select the 'Bitmap-Embed' item.

## Dropping Template Code

---

Movicon provides you with the possibility to edit basic routines which are to be executed when inserting Templates on Screen. Events will be made available internal these routines and automatically executed by Movicon while the Template is being inserted. These events are:

- **OnCustomizeSimbol()**
- **OnVariableCreating**
- **OnVariableCreated**

You can access this basic code from the **"Symbols Library"** by opening the **"Dynamic Property Inspector"** window of the Symbol in question. By opening the **"Dynamic Property Inspector"** of any one of the Templates within the Library, the Dropping Code tab will appear as the first item in the symbol's list of components. When selecting this item, where the variables and codes of each of the Symbol's components normally appear, the script code will appear, being the event routines listed above. Code can be inserted within in these routines in order to customize the Template. This functionality is very handy for setting the static properties of symbols. This can be done by using the **Prop()** function from the **DrawCmdTarget Interface** to set those static properties concerned in the symbol. When these properties are set then inserting the Template using the code contained in the Dropping Code, they will remain permanent in the symbol when

saved together with that symbols characteristics, thus becoming the symbols unaccessible constants that cannot be changed through the symbol's properties.  
 The Dropping Code is useful for customizing Templates being inserted from the Library. Once the Templates has been inserted onto the screen you can Drop Code by "Shift + Double-Clicking" on the Template or by using the "Open..." button in the Template's Properties Window. This will allow you to modify the Template's properties later on as well.

#### OnCustomizeSymbol() Event

This event is executed when the Dropping Code is called, therefore while a Template is being inserted on Screen from the Symbol Library or by "Shift + Double-Clicking" on the Template after it has been inserted on Screen.

This event has the following declaration:

```
Sub OnCustomizeSymbol(bRet As Boolean, bShowPropInsp As Boolean,
bCreateVariables As Boolean)
```

```
End Sub
```

This event's parameters are:

Parameters	Description
bRet as boolean	When set at False, the Template insertion operation will be cancelled (default = True)
bShowPropInsp as boolean	When set at False, the 'Dynamic Explorer Property' window will not display after the symbol has been inserted
bCreateVariables as Boolean	When set at False, the Template's variables will not be inserted in the Real Time DB (default = True)

#### Event OnVariableCreating()

This event is called each time a variable must be added to the project when a symbol is being inserted. In this way you can executed controls more accurately on variables which must be inserted with the template. This event has the following declaration:

```
Sub OnVariableCreating(bRet As Boolean, VariableName As String,
bCreateLocal As Boolean)
```

```
End Sub
```

The parameters of this event are:

Parameter	Description
bRet as boolean	When set at False, it cancels the operation to insert the variable (default = True).
VariableName As String	Contains the name of the variable to be added to the project. The value of this parameter can be changed. In this way a variable will be inserted with a new name inserted. When the variable identified by the "VariableName" parameter already exists, then a new variable will be added with the same name but with a progressive final number.
bCreateLocal As Boolean	When set at True, the variable will be created locally to the screen. (default = False. When the symbol is inserted with the shift pressed: default = True).

#### OnVariableCreated() Event

This event is executed each time a variable is created in the project while a symbol is being inserted. By using this event you can modify the properties (only those in read/write) of variables which are added to the project while a Template is being inserted. This event has the following declaration:

Sub **OnVariableCreated**(VariableName As String, bCreateLocal As Boolean)

End Sub

The parameters of this event are:

Parameter		Description
VariableName String	As	Contains the name of the variable which has been created in the project.
bCreateLocal Boolean	As	When the parameter's valued is True this means that the variable has been created as a local screen variable, otherwise when False this means that the variable has been created in the RealTime DB.

## 9. Draws and objects

---

***The objects which can be inserted into screens are all vectorial type objects and all their properties can be configured and modified.***

The Movicon Controls and Objects which can be inserted in to Screens are available from the "**Objects Window**". These components can carry out different functions, from simple geometric shapes to advanced controls for command execution or displaying data.

The Movicon Objects and Controls are created in proprietary vectorial format, and symbols (composed of more components) can also be created and animation properties associated to them. This format can be exported or imported from the Metafile format (WMF,EMF).

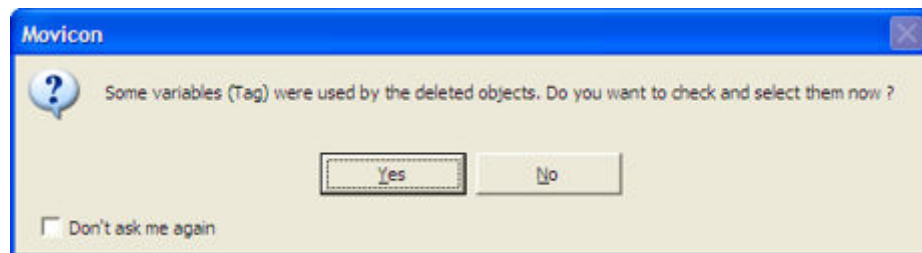
**Even though various components carry out different functionalities, some of their properties, in particular all the properties concerning graphic animations, are the same for all and are available for each object.**

### Deleting Objects from Screens

---

When an object or symbol, to which variables have been associated (ie. in their "Variables" or "Animations" properties) is removed from the screen a message window will open showing this text:

**Some Variables (Tag) were used by the deleted object. Do you want to check or select them now?**



When confirming with "Yes" the "List Variables (Tags) resource will open in the Project Explorer window with the interested variables highlighted (selected). At this point the programmer can decide which to delete or keep the variables as they are.

### Tooltips in Controls

---

Each control that can be inserted on screen shows a Tooltip for default when the mouse pointer is positioned on it. This Tooltip appears both in the project development and Runtime modes. During the Runtime mode this Tooltip shows information inherent to any variable associated to the object (name, value, quality, etc.). However, in development mode, the tooltip shows a list of variables associated to the object (if any) along with some information inherent to the object itself, such as how to access the object's properties etc. For instance, in the Trend object the tooltip explains how to add pens and so forth.

The tooltip can be disabled from displaying in Runtime by simply disabling the "Show Dynamic Tooltip" property from each objects' 'General' property group. The Tooltip will always remain active in development mode.

# Basic Shapes

---

The Movicon basic shapes which can be inserted on screen can be selected from the "**Objects Window**". These components are mainly simple geometric shapes which can be graphically animated.

The objects available in this class are as follows:

## Line

This object is a simple vectorial line. After activating the command, click on any point you wish the line to start from, then drag the mouse until you reach the point you wish to end the line with.

By doing this you should finish with a straight line whose style can be configured in the "Properties Window". Its position can be changed by re-selecting it and dragging with the mouse as please. To change its sizes, select one end and drag it with the mouse until you get the size desired.

## Rectangle

This object represents a vectorial square or rectangle shape. After activating the command, click on any point corresponding to the top left corner for rectangles, then drag the mouse until you get the size desired. You can then configure the rectangle's style by using the "Properties Window". You can change its position by re-selecting it and dragging it to the new point desired. To change its size, just select the small squares on its borders and drag to the new size desired.

## Rounded Rectangle

This object represents a vectorial square or rectangle shape with rounded corners. After activating the command, click on any point corresponding to the top left corner for rectangles, then drag the mouse until you get the size desired. You can then configure the rectangle's style by using the "Properties Window". You can change its position by re-selecting it and dragging it to the new point desired. To change its size, just select the small squares on its borders and drag to the new size desired.

## Ellipse

This object represents a circular or elliptic vectorial shape. After activating the command, click on the desired point corresponding to the top left quadrant for circular shapes, then drag it with the mouse until you reach the point desired. You can then configure its style through the "Property Windows". Its position can be changed by re-selecting and dragging it to the position desired with the mouse. To change its sizes, just select the small squares on its border and drag it to the size desired with the mouse.

## Polygon

This object represents a geometric shape made up of lines. After activating the command, click on the point desired and, by moving the mouse, each time a click is made a vertex will be added to the polygon. Movicon will create a vectorial geometric shape with lines by forming together all the vertexes which have been set. The start and end points will always be automatically connected together by a vertex line.

When the **ESC** key is pressed to terminate the drawing procedures, the last vertex proposed by Movicon will be kept even though not executed with a click.

When the **ENTER** key is pressed to terminate the drawing procedures, only the vertexes confirmed with the mouse click will be kept and not the last one proposed by Movicon.

After this has been done you can configure its style through its "Properties Window". You can change its on screen position by selecting it and dragging it to where you please with the mouse.

To change its shape or sizes, select the small squares which represent the polygon's vertexes and drag them with the mouse until you get the new shape desired.

## Polyline

This object represents a geometric shape made up with a series of contiguous straight vectorial lines. After activating the command, click on the point desired and mouse it with

the mouse. Every time you click a vertex will be added to the object where the cursor is positioned. Movicon will create a geometric vectorial shape made from contiguous lines of the vertexes which were set. The start and end points will not be connected together.

When the **ESC** key is pressed to terminate the drawing procedures, the last vertex proposed by Movicon will be kept even though not executed with a click.

When the **ENTER** key is pressed to terminate the drawing procedures, only the vertexes confirmed with the mouse click will be kept and not the last one proposed by Movicon.

After this has been done you can configure its style through its "Properties Window". You can change its on-screen position by selecting and dragging it to where you please with the mouse.

To change its shape or sizes, select the small squares which represent the Polyline's vertexes and drag them with the mouse until you get the new shape desired.

### **Polybezier**

This object represents a geometric shape made up with a series of contiguous curved vectorial lines. The curve is formed by two intersections between the start and start point. After activating the command, click on the point desired and mouse it with the mouse. Every time you click a vertex will be added to the object where the cursor is positioned. Movicon will create a geometric vectorial shape made from contiguous lines of the vertexes which were set. The start and end points will not be connected together.

When the **ESC** key is pressed to terminate the drawing procedures, the last vertex proposed by Movicon will be kept even though not executed with a click.

When the **ENTER** key is pressed to terminate the drawing procedures, only the vertexes confirmed with the mouse click will be kept and not the last one proposed by Movicon.



The Polybezier must be composed of four points, one start point, two intersection points and an end point.

After this has been done you can configure its style through its "Properties Window". You can change its on-screen position by selecting and dragging it to where you please with the mouse.

To change the curved corner, select the small squares outside the curve and drag it with the mouse until you get the shape desired.

To change the start or end point, select and drag the small squares placed at both ends of the curve.

### **Arc**

This object represents a semicircular or semi-elliptic vectorial line. After activating the command, click on the point desired and drag it with the mouse until you reach the end point desired. You can then configure its style through the "Property windows".

To change the curve, select the small filled in squares on the border and drag them with the mouse until you get the size and arc desired.

To change the arc's start or end point, select and drag the corresponding small empty squares along the border.

### **Cord**

This object represents a portion of a vectorial semicircle or elliptic. After command activation, click on the point desired, then drag the mouse until you reach the end point desired. Movicon will unite both ends of the circle portion with a straight line after which its style can be configured through its "Properties Window". You can change its position by selecting and dragging it to a another position.

To change its curve angle, select the filled in small squares on the border and drag them with the mouse until you get the size and angle desired.

To change cords's start and end point, select and drag the small empty squares placed inside the object.

### **Pie**

This object represents a portion of a vectorial semicircle or elliptic. After command activation, click on the point desired, then drag the mouse until you reach the end point desired. Movicon will unite both ends of the circle with the centre of the ellipse you have drawn. After this you can configure its style through the "Properties Window". You can change its position by selecting and dragging it to a another position.

To change its curve angle, select the filled in small squares on the border and drag them with the mouse until you get the size and angle desired.



To change cords's start and end point, select and drag the small empty squares placed inside the object.

#### Text

This object represents a text object. After command activation, click on the point desired and drag it until you reach the end point desired. Movicon will display a default text inside which you can replace, along with the font, after you have activated its "Properties Window".

The Text object's properties allow you to set it with a fixed size or to make it adaptable to the object's sizes.

You can change its position by selection and dragging it with the mouse to another position.

You change it object's sizes, by selecting and dragging the small squares on its borders with the mouse until you reach the size desired.

#### Square

This object represents a vectorial square shape. This shape is originally a "Rectangle" with its **"Preserve Aspect Ratio"** enabled so that when it is resized it will always keep its sides in proportion to form a square and not a rectangle.

#### Circle

This object represents a vectorial circle shape. This shape is originally an "Ellipse" object with its **"Preserve Aspect Ratio" properties enabled so that when its is resized it will always keep its circle shape and not the shape of an ellipse.**

## Object Editing Command Shortcuts

---

This paragraph is a detailed summary on a list of all the commands which are used to speed up graphic interface editing operations.

You may find that some of these commands are not part of your daily use of this product even though indicated in the descriptions of each single object. The detailed list in this recap will give you a better idea on how to use of these shortcuts to the full in order to perform your tasks quicker.

### DRAG & DROP

---

The Drag & Drop technique, a standard widely used in Windows, permits you to drag and drop objects or resources to different locations according to the task one has been set with.

This technique has been used in Movicon to allow you to obtain associations between variables, Data Logger, Schedulers and objects, and object quickly, as well as the usual dragging of objects and resource to modify the project structure.

Object	DRAG & DROP Variable operations
Trends Charts Display/Spin ListBox Gauge/Meters Buttons Hot Region	To assign a variable to an object without going through the object's properties manually, Just drag & drop the variable (click and keep the mouse key pressed down on the desired variable in the "Variable List" from the Project Explorer and drag it to the location desired and then drop it by releasing mouse key) onto an object on screen. <b>The variable assigned in this way will become the object's reference variable.</b>
Trends	Dragging a variable (Tag) from the Variable List resource directly onto a Trend will get it assigned to that Trend. The drag&drop works in the same way as above but the pens are removed from the trends and are recreated according to the number of variables dragged and dropped into the trend.

Object	DRAG & DROP Data Logger operations
Trend Charts Data Logger/Recipes Windows	You only need to drag&drop a Data Logger (click and keep mouse key pressed down on the Data Logger, from the Data Logger and Recipes' resource in the Project Explorer, and drag it) on to an object on screen to assign it to that object, saving you time in having to do it manually through the object's properties. <b>The Data Logger assigned in this way will become the object's reference Data Logger.</b>

Object	DRAG & DROP Scheduler Operations
Scheduler Window	You only need to drag&drop a Scheduler (click and keep mouse key pressed down on the Scheduler, from the 'Scheduler Object List' resource in the Project Explorer, and drag it) on to an object on screen to assign it to that object, saving you time in having to do it manually through the object's properties. <b>The Scheduler assigned in this way will become the object's reference scheduler.</b>

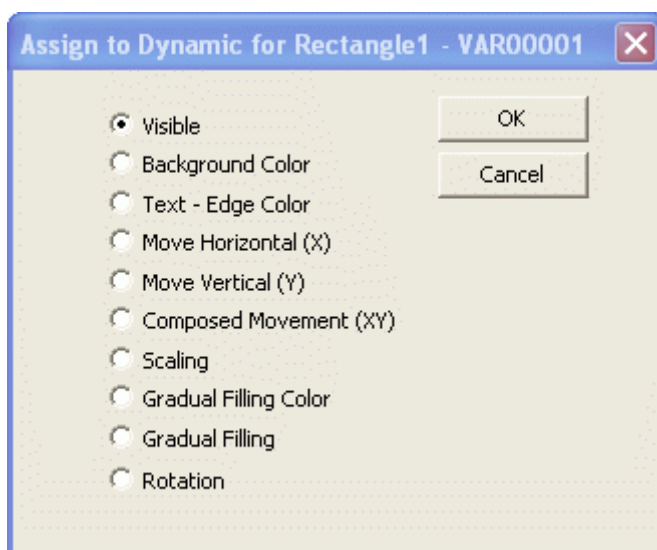
You can also **create new objects** on screen associating them directly to the variable dragged on screen. In order to do this you need to:

1. select the object you wish to create from the toolbox (eg. Display)
2. select the desired variable from the Project Explorer's Variable List
3. Drag and drop the variable to an empty area in the screen
4. release the mouse key and Movicon will create the object and assign the variable



**During the Drag & Drop operations the mouse icon will change image. If the resource is dragged onto an object that cannot be assigned with that resource, the mouse icon will change to the 'no entry' symbol. For example, this will happen when trying to drag a Data Logger onto a "Button" object.**

if a variable is dragged onto an object and the object itself has no specific property for variable assignment, a dialog window is open in order to select the object's animation property to assign the variable to:



If a variable is dragged onto a composed symbol, keeping the "SHIFT" key pressed will cause the variable to be assigned to the container object (that is, to the symbol itself), otherwise the variable is assigned to the symbol component where the mouse cursor is placed.

### Drag & Drop into Templates

Object	DRAG & DROP operation
Alarms (Templates)	Dragging and dropping an Alarm object (from the project's Alarms resource) on a variable will assign the alarm to the variable as a template.
Data Loggers (Templates)	Dragging and dropping a Data Logger object (from the project's DataLogger/Recipe resource) on a variable will assign it to the variable as a template.
Events (Templates)	Dragging and dropping an Event object (from the project's Event object resource) on a variable will assign it to the variable as a template.

### SHIFT + DOUBLE CLICK

Some of the Movicon Objects and Controls that can be inserted into Screens, available in the "**Objects Window**", can be configured through the supplementary setting windows as well as the "Properties Windows". To access these setting windows you need to:

- Press the Shift key and double left mouse click on the object (Shift+DbIClick)
- Press the "General" group's "Open" button in the object's "Properties Window"
- Click on the small circle which appears on top right of the object when selected

This table shows the setting window which will open according to the control selected:

Object	SHIFT + DOUBLE CLICK result
Buttons and Hot Region	The "Command List" settings window is open where a list of commands to be executed can be associated to the control.
Chart	Chart's configuration window is opened.
Trend	The Trend Pen Configurations window is opened.
Grid	The window for selecting the 'data source DSN to be associated to the grid is opened.
Alarm Window	The "Field Choice" window is opened for setting the columns to be displayed in the Alarm Window.
Historical Log Window	The "Field Choice" window is opened for setting the columns to be displayed in the Historical Log Window.
DataLogger/Recipe Window	The "Field Choice" window is opened for setting the columns to be displayed in the DataLogger/Recipe Window.

DB Trace Window	The "Field Choice" window is opened for setting the columns to be displayed in the DB Trace Window.
Embedded Screen	A window opens for selecting the Screen to be associated to the object. <b>Caution:</b> for Embedded Screens clicking on the small circle which appears on the top right of the object will activate the open command in the workspace of the screen associated to the object.

### "Dynamic Properties Explorer" window

The "SHIFT + Click" command on the small circle appearing on the top-right corner of the selected object opens the "Dynamic Property Inspector" window of the symbol itself.

## SHIFT + CTRL + DOUBLE CLICK

A very quick way for setting the object's main settings is to use the simplified properties POP-UP Window of an object. This window only contains the main properties without displaying the whole list of properties.

To activate this window you need to double click while keeping the Shift and CTRL keys pressed down or click on the object's top right corner. To deactivate this window just click on any point of the screen.

If you use this window, you will simplify and speedup your routine work. the listed properties have been further simplified and reduce in respect to those in the "Easy Mode" according to the description in the initial paragraph relating to "Properties Windows".

## CTRL+ DOUBLE CLICK

The CTRL+double click command activated the explorer window of an object's VBA script.

## Open applications using the "Browse File" button

Object properties which allow file names to be entered using the "Browse File" (...) button, such as image files or an object configuration file, permit you to use this same button (...) to launch any application associated to the file extension. This can be done by simply clicking on the browse button while keeping the "CTRL" key pressed down. For instance, Screen Background Image properties permit you to select a background image. If you choose a "bmp" image, and this extension is associated to a "MS Paint application when pressing the CTRL key while clicking the (...) button will open the "MS Paint" application with the "bmp" image in edit mode in a new window. This mechanism will only work if a file has already been selected in the object's property and the file's extension has been associated to an application. The application window can be opened in minimize mode.

The properties which have been equipped with this kind of function are:

- **Screens:** Background - Background image
- **Drawings and Symbols:** Background - Static Image
- **Button Objects:** Background - Released Button Image, Pressed Button Image, Selected Button Image
- **Grid Objects:** Execution - Text File
- **Drawings and Symbols:** Execution - Configuration File

## Symbol Configuration Files

Each control or composed symbol, that can be inserted on screen, may be associated with a Configuration File to save their settings in order to be reactivated in both the design/programming or Runtime phases. The file will be saved in XML xml format with the ".xml" extension. Some execution properties have been provided for each control in order to setup its Configuration File:

- **Ext. File Settings**
- **Generate Unique File name**
- **Load**
- **Save**

The Configuration File settings are loaded in design mode with the "Load" command, but are loaded automatically when screens are loaded in Runtime mode.

Only some of the settings saved on file are restored when using the "Load" command (in design mode) or when screens are being loaded (in Runtime mode). Mostly all the properties are restored except those regarding the object's positions (Position X, Position Y, Width, Height).

To avoid XML files sizes being too big, many of the properties are saved only if different from their default values.



Properties which don't get saved in the "sxml" file are those which have a default value, and therefore will not be restored when the file gets loaded. For instance, when a "sxml" file is saved with some properties that still have their default values, which afterwards are changed and reloaded before the file is saved again, the default values for these properties will not be restored.



To display the restored properties in design mode using the "Load" command in the properties window, you will need to deselect and reselect the object.

During the Runtime phase you can save and restore configurations using the "SaveExtSettings", "LoadExtSettings" and "ExtSettingsFile" basic script functions. **These functions are currently available for the "TrendCmdTarget" interface only and therefore for the "Trend" object only.**

## Object Default Settings File

Movicon consents the programmer to customize those Objects contained in the "ToolBox Window" allowing the programmer to modify characteristics of simple Objects without having to create completely new symbols and adding them to the Symbol Library. This can be done by creating a "ObjectName.defxml" file for each object to contain their default settings. Once you have done this, every time you insert one of these objects on screen from the Toolbox, it will already have the settings predefined in its file. These files contain tags and attributes specifically for the objects in question and have the same format with which the object is defined with on screen. The opening tag must be <default>.

For instance, the "rectangle.defxml" file shown below will be the one used by Movicon when a "Rectangle", is inserted on screen:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<default>
<type x="300" y="220" width="-250" height="-120" fill="rgb(255,0,0)"
stroke="syscolor(WINDOWTEXT SysColor)"
stroke-width="1" stroke-linecap="butt" stroke-linejoin="round">rect</type>
<ExtFill fill="syscolor(WINDOW SysColor)">0</ExtFill>
<Name acr="4294901760"/>
<text font-family="Tahoma" font-size="8" text-anchor="middle" font-weight="400"
pf="34" f3D="0"/>
<Shadow>0</Shadow>
<Rotation>0</Rotation>
</default>
```

By using this file, the inserted "Rectangle" object will have a red background color according to the modified "fill" tag. To create settings files for objects you can use this procedure as an example:

1. insert the desired object on the screen, the rectangle for instance, and modify the properties of interest (ie. the border, char font, etc.)

2. select the object, copy it to the window clipboard (Ctrl+C) and then paste it in a text editor such as Note Pad
3. cancel the <MovClipboard>, <child> and <object> Tags in the xml text shown in the text editor
4. then add the title with the xml version and Tag <default> to the code
5. save the file with the "Rectangle.defxml" name

The initial code that is copied in the text editor would have been as follows:

```
<MovClipboard>
<child>
<object>base</object>
<type x="290" y="170" width="-190" height="-120" fill="rgb(255,0,0)"
stroke="syscolor(WINDOWTEXT SysColor)" stroke-width="1" stroke-linecap="butt"
stroke-linejoin="round">rect</type>
<ExtFill gcolor="rgb(0,0,0)" Style="3" fill="syscolor(WINDOW SysColor)">0</ExtFill>
<Name acr="4294901760"/>
<text font-family="Tahoma" font-size="8" text-anchor="middle" font-weight="400"
pf="34" f3D="0"/>
<Shadow>0</Shadow>
<Rotation>0</Rotation>
</child>
</MovClipboard>
```

In order to use as a settings file, the code would then be modified as follows:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<default>
<type x="290" y="170" width="-190" height="-120" fill="rgb(255,0,0)"
stroke="syscolor(WINDOWTEXT SysColor)" stroke-width="1" stroke-linecap="butt"
stroke-linejoin="round">rect</type>
<ExtFill gcolor="rgb(0,0,0)" Style="3" fill="syscolor(WINDOW SysColor)">0</ExtFill>
<Name acr="4294901760"/>
<text font-family="Tahoma" font-size="8" text-anchor="middle" font-weight="400"
pf="34" f3D="0"/>
<Shadow>0</Shadow>
<Rotation>0</Rotation>
</default>
```

Any created setting files must then either reside in the specific project "IMAGES" folder or in the Movicon installation folder.

Modifications to the object's default sizes when being inserted on screen by the user have not been provided for. Setting the "width" and "height" attributes from the ".defxml" file to value "0" will cause the following behaviour when inserting the object on screen:

- a simple mouse click on the screen will add the object with the default sizes established by Movicon
- the object can be resized by clicking and keeping mouse button pressed down to move the object to the size desired with the arrow starting from 0x0.



The "width" and "height" attributes have been already set at zero in the ".defxml" files which Movicon Setup installs.

As already mentions, the file names are taken as default names of the object categories depending on the Movicon installation language used. For some objects, such as Buttons and meters, the filed name follows a different logic, as explained ahead. Furthermore, you must keep in mind that each object, such as the Rectangle and Text, take their settings from the same file, therefore the same customization will effect both objects.

This table lists the files names to be used for different objects:

Object Name	File Name (Italian)	File Name (English)
Line	Linea.defxml	Line.defxml
Rectangle	Rectangle.defxml	Rectangle.defxml
Rounded Rectangle	Rettangolo Smussato.defxml	Rounded Rectangle.defxml
Ellipse	Ellisse.defxml	Ellipse.defxml
Polygon	non previsto	not provided
Polyline	non previsto	not provided
Pipe	non previsto	not provided
Polybezier	non previsto	not provided
Arc	Arco.defxml	Arc.defxml
Chord	Corda.defxml	Chord.defxml
Pie	Torta.defxml	Pie.defxml
Text	Rettangolo.defxml	Rectangle.defxml
Square	Rettangolo.defxml	Rectangle.defxml
Circle	Rettangolo.defxml	Rectangle.defxml
Horizontal Plotter	Trend.defxml	Trend.defxml
Vertical Plotter	Trend.defxml	Trend.defxml
Horizontal Trend	Trend.defxml	Trend.defxml
Vertical Trend	Trend.defxml	Trend.defxml
Trend XY	Trend.defxml	Trend.defxml
Data Analysis	Data Analysis.defxml	Data Analysis.defxml
Chart	Grafico.defxml	Chart.defxml
Grid	Griglia.defxml	Grid.defxml
List Box	Lista.defxml	List Box.defxml
Group Box	3_0_Group Box.defxml	3_0_Group Box.defxml
Embedded Screen	Sinottico Incastrato.defxml	Embedded Screen.defxml
Horizontal Spin	non previsto	not provided
Vertical Spin	non previsto	not provided
Display	Display.defxml	EditBox-Display.defxml
Tab Group	Gruppo Tab.defxml	Tab Group.defxml
Combo Box	Combo Box.defxml	Combo Box.defxml

ActiveX	non previsto	not provided
OLE Object	non previsto	not provided
Alarm Window	Finestra Allarmi.defxml	Alarm Window.defxml
Log Window	Finestra Log Storico.defxml	Log Window.defxml
Data Logger-Recipe Window	Finestra Data Logger-Ricette.defxml	Data Logger-Recipe Window.defxml
TraceDB Window	Finestra DB Trace.defxml	TraceDB Window.defxml
Scheduler Window	Finestra Schedulatore.defxml	Scheduler Window.defxml
IP Camera Viewer	Finestra Telecamera IP.defxml	IP Camera Viewer.defxml

Setting file names for "Button" type objects are composed more precisely:

<#object kind>\_<#pushstyle>\_category.defxml

where:

<#object kind> = object type. Possible values are:

- 0 = button
- 1 = check box
- 2 = option box
- 3 = group box
- 4 = hot region

<#pushstyle>: corresponds to the "pushstyle" attribute of the xml code relating to the object

category: identifies the object category. The value is "Button"

In this case the settings file name does not depend on the Movicon installation language but will always be composed in the same way:

Object Name	File Name
Pointer	1_0_Check Box.defxml
Checkbox	2_0_Radio Button.defxml
Hot Region	4_0_Hot Region.defxml
Push Buttons	0_0_Button.defxml
3D Buttons	0_1_Button.defxml
Yellow Light	0_2_Button.defxml
Blue Light	0_3_Button.defxml
Green Light	0_4_Button.defxml
Red Light	0_5_Button.defxml
Yellow Button	0_6_Button.defxml
Blue Button	0_7_Button.defxml
Green Button	0_8_Button.defxml



Red Button	0_9_Button.defxml
Emergency	0_10_Button.defxml
Switch	0_12_Button.defxml
Selector A	0_13_Button.defxml
Selector B	0_14_Button.defxml
Square Blue	0_15_Button.defxml
Square Red	0_16_Button.defxml
Square Yellow	0_17_Button.defxml
Square Green	0_18_Button.defxml
Square Black	0_19_Button.defxml



Radio and Check buttons do not inherit back colours from the ".defxml" file.

The settings file name for "Meter" type objects is composed more precisely:

<#type>\_<#knobstyle>\_category.defxml

where:

<#type>: represents the Meter type, such as "horizontal", "vertical" or "round", and corresponds to the "style" attribute in the object's xml file

<#knobstyle>: represents the "Cursor Style" and corresponds to the "Knobstyle" attribute in the object's xml file

category: identifies the object category. The value is "Gauge"

In this case the settings file name does not depend on the Movicon installation language but will always be composed in the same way:

Object Name	File Name
Vertical Slider	0_3_Gauge.defxml
Horizontal Slider	1_3_Gauge.defxml
Gauge	2_3_Gauge.defxml
Top Gauge	2_3_Gauge.defxml
Bottom Gauge	2_3_Gauge.defxml
Left Gauge	2_3_Gauge.defxml
Right Gauge	2_3_Gauge.defxml
Meter	2_3_Gauge.defxml
Top Meter	2_3_Gauge.defxml
Bottom Meter	2_3_Gauge.defxml
Left Meter	2_3_Gauge.defxml

Right Meter	2_3_Gauge.defxml
Vertical Bargraph	0_3_Gauge.defxml
Horizontal Bargraph	1_3_Gauge.defxml
Vertical Bargraph Grid	0_3_Gauge.defxml
Horizontal Bargraph Grid	1_3_Gauge.defxml
Dundas Gauges	not provided

## Outside Screen Objects

---

While in Design mode you can verify situations where one or more objects have been placed completely outside the screens they belong to. These objects will remain visible in the area outside the screen and can be modified in the usual way by selecting them to access their properties through the Properties Window. However, objects outside screens will not be made visible in project runtime mode, and those partly kept on screen will be shown with the part included on the screen only.

There are different reasons causing situations to show objects outside their screens:

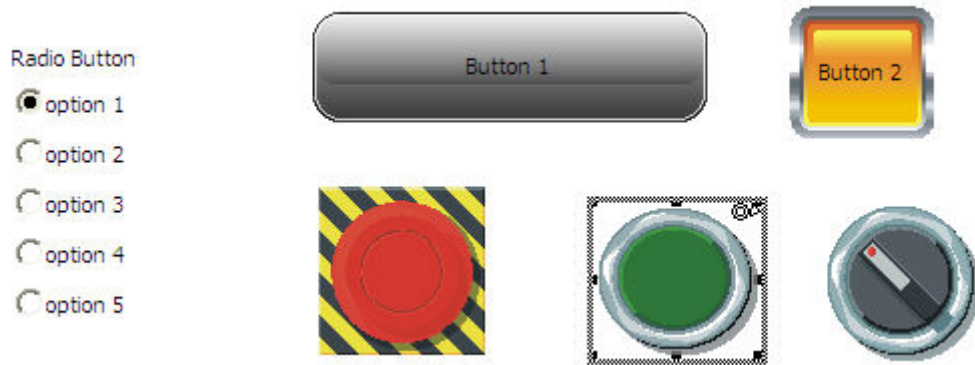
- in screen edit mode where the object is dragged outside screen
- in screen edit mode when the object's X or Y Position properties have been inserted with external screen values
- in screen edit mode copying an object from one screen to a smaller one, where it can't fit in entirely and therefore remains outside
- in screen edit mode where the screen is reduced in size without adapting the objects it contains (see "Screen resize")

When selecting the screen in the "Refactoring Explorer" window, a list of the objects showing completely outside the screen will display. In this case, the following message will show:

The 'Object Name' is not in the screen boundary and won't be visible in runtime"

# Buttons and Hot Regions

The Movicon Buttons and Hot Regions are elements through which Commands can be executed or project variables can be set. They are accessed from 'Buttons-Lights-Switches' group found in the **"Toolbox Window"**.



## Check Box Buttons

This is a typical two state button. A Movicon variable can be associated to this control and set at the '0' value (unchecked box) or '1' (checked box) according to the box's status. The status is selected by mouse clicking or by using the TAB or SHIFT+TAB keys and pressing ENTER or the Space Bar on the keyboard.

## Radio Button

This is a common exclusive option button. You can associate a Movicon variable to this control which will assume the progressive numeric value based on the box selected. For instance, when the first option box is selected the variable will assume the '0' value, when selecting the second option box the variable will assume the '1' value, then '2' and so forth. The number of options to be displayed is set through the object's properties. The selection is done by Mouse clicking or using the keyboard by selecting the component with the TAB or SHIFT+TAB keys and pressing the UP/DOWN ARROW keys.

## Buttons

The Movicon Buttons are objects of different graphically built shapes that consent command activation when activated with the mouse or keyboard keys during project processing.

The Buttons can be associated with the commands available in the Movicon **"Command List"** window.

The buttons' shapes or styles can be setup through the **'Properties Window'** by selecting the one desired from those already created.

The execution associated to the command is activated by clicking on the button or using the keyboard where the TAB or SHIFT+TAB keys are used to select the command and then ENTER or the Space Bar keys are used to activate it.



There are some controls, such as lights, illuminated buttons, switches and selectors available in the 'Buttons-Lights-Switches' group in the **"Objects Window"**. All these controls derive from the Button objects by simply using the control's Style and Execution properties.

## Hot Regions

The Hot Regions are built from zones inserted in strategic points in the Screen which are invisible but mouse sensitive during project Runtime.

Since the Hot are invisible they do not have style properties but only one execution property for setting the associated command for when the operator clicks on Hot Region area.

When inserting a Hot Region in the Screen, this will appear as a shaded area with default sizes. After inserting the Hot Region it can be sized as pleased by dragging the borders with the mouse.



The Screen zone within which the Hot Region has been placed, it built only with a background representing the 'strategic' design. During project processing this zone will not be visible, but when the mouse pointer goes into or passes over the Hot Region area, the borders of this area will highlight. The programmer can also setup a 'ToolTip' text to appear if desired.

The execution of the command associated to the Hot Region is activated by clicking the mouse or using the keyboard by selecting the Hot Region with TAB or SHIFT+TAB keys and pressing ENTER or Space Bar.

## Components deriving from Button Objects

---

The Button object can take on different graphical shapes and different functions according to how it is configured. Some of these variations are already available in the **"Buttons-Lights-Switches"** from the **"Objects Window"**.

### 3D Buttons

The 3D button is a normal button changed in the following properties:

- "Style - Style": changes its graphical aspect only, by giving it a round 3D look.

### Lights (colored)

The "Light" objects that can be inserted are normal buttons changed in the following properties:

- "Style - Style": changes its graphical aspect only, by making it look like a luminous indicator.
- "Style - Clickable": this property is disabled when the light is for display only
- "Execution - Mechanic Style": this property is enabled for managing the mechanic "ON-OFF" which sets Light's modality with an On or Off status.

### Buttons (colored)

The colored "Button" objects which can be inserted are normal buttons changed in the following properties:

- "Style - Style": changes its graphical aspect to look like a round switchboard button.
- "Execution - Mechanic Style": this property is enabled for managing the mechanic "ON-OFF" to make an ON/OFF command button type.

### Emergency

The "Emergency" objects which can be inserted are normal buttons changed in the following properties:

- "Style - Style": changes only its graphical aspect to look like an emergency switchboard button.
- "Execution - Mechanic Style": this property is enabled for managing the mechanic "ON-OFF" to make an ON/OFF command button type.

### Switch

The "Switch" objects which can be inserted are normal buttons changed in the following properties:

- "Style - Style": changes its graphical aspect by making it look like a plant ON/OFF switch.
- "Execution - Mechanic Style": this property is enabled for managing the mechanic "ON-OFF" to make an ON/OFF command button type.

### Selector A/B

The "Selector A/B" objects which can be inserted to are normal buttons changed in the following properties:

- "Style - Style": changes its graphical aspect to make it look like a ON/OFF switchboard selector switch.
- "Execution - Mechanic Style": this property is enabled for managing the mechanic "ON-OFF" to make an ON/OFF command button type.

### Square (coloured)

The coloured "Squares" that can be inserted are normal buttons which have been modified in the following properties:

- "Style - Style": changes its graphical aspect only to look like a square button found on control panels.
- "Execution - Mechanic Style": The "ON-OFF" mechanics is enables to make it an ON/OFF command button type.

## Button and Hot Region Execution Properties

---

The Execution properties of Buttons and Hot Regions allow you to set the type of command which the control must execute.

To edit the Execution properties, select the object with the mouse and then use the Movicon **'Properties Window'**.

The pushbutton objects executed two different actions types:

1. Direct action on a Tag
2. Command list execution

Actions are defined and configured depending from the "Mechanic Style" assigned to the button, as following described.

### Check Variable

The name of the variable to be used as the check variable, which will be managed differently according to the command type selected in the "Mechanic Style" property, is entered in this editbox (or selected by using the '...' browse button on the right).

This property is only provided for "Buttons", **"Check Box Buttons"** and **"Radio Button"**.



The Check Variable can be configured with any kind of "Mechanic Style". If the style chosen is "Command Execution", the Check Variable allows to define the status of the button (pressed or released) or the colour for the light buttons.

### Mechanic Style

Using this selection will be chosen the operating mode of the push button.

<b>ON-OFF</b>	The button will act as On-Off on the <b>"Check Variable"</b> property assigned variable.
<b>Impulsive</b>	The button will act as a pulse on the <b>"Check Variable"</b> property assigned variable. It means that the variable will go to the "1" state for the time defined in the property <b>"Impulsive Time"</b> .
<b>Execute Commands</b>	The button will execute the command or the commands list defined. The command execution could be executed on the button pression, on the button release or while down. In this last case, the execution could be cyclic, repeated depending from the time period defined from the property <b>"Impulsive Time"</b> .

### Impulsive Time

The property "Impulsive Time" can be used for two different scope:

1. Using the Mechanic Style "Impulsive" you can define the time (msec) for the pulse on the check variable. If you set this time to zero, the variable will be put to "ON" until the button is pressed.
2. Using the Mechanic Style "Execute Command" you can define the cyclic execution (msec) for the commands defined as ""Command while down". **In this case the time pulse as "zero" is not allowed and it will disable this function.**

### Command on Release

This button opens the Movicon '**Command List**' window through which you can setup a list of one or more commands which will be executed **upon the button release event**.

This property is only available for 'Button' and 'Hot Region' objects.

For further information on the commands available please refer to the paragraph on "**Command List**".

### Command While Down

This button opens the Movicon '**Command List**' window through which you can setup a list of one or more commands which will be executed **while the button kept pressed down**. **In this case, commands can be executed several time (cyclically) until the button is pressed, with a period time defined by the property "Impulsive time"**.

This property is only available for 'Button' and 'Hot Region' objects.

For further information on the commands available please refer to the paragraph on "**Command List**".

### Command on Pressed

This button opens the Movicon '**Command List**' window through which you can setup a list of one or more commands which will be executed **upon the button pressed event**.

This property is only available for 'Button' and 'Hot Region' objects.

For further information on the commands available please refer to the paragraph on "**Command List**".

### Command on Mouse Down

When this property is enabled, the Command List associated to the button will be executed upon the Mouse Down event (when the mouse button is pressed down) instead of the Mouse Up event (when the mouse button is released).



This option is presented only for command buttons (being those which have been associated with a Command List) and not for selection or impulsive buttons.

### Num. Radio Buttons

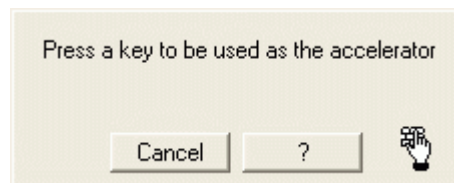
The number of options buttons to appear for the control type 'Option Buttons' is entered in this edit box.

### Enable Shortcut

When enabling this check-box, an Shortcut key will be associated to the control. Therefore the control's command can be executed during runtime by simply pressing the preset Shortcut key (eg. 'F1'). After having activated this property it will be necessary to setup the Shortcut key to be used through the '**Shortcuts**' property.

### Shortcut

The key or combination keys you wish to associate to the control can be entered in this edit box. Movicon offers you the possibility to automatically enter the key to be associated by pressing the key directly on the keyboard. In order to do this press the '...' button on the right hand side of the entry box to display the following window:



At this point just press any one of the buttons on the keyboard, or a combination of keys also consisting of the 'modifier' keys, to register them in the 'Shortcut' box. This operation can be cancelled with the 'Cancel' button.



The Shortcut key is only active when the control's 'Enable Shortcut' property has also been enabled.

#### **Show Shortcut**

Enabling this check-box will show the selected 'Shortcut' key at the side of the control's title.

## **Button Background Attributes Properties**

---

The Button Background attributes consent you to associate images to the button object. To modify the Background properties, select the object with the mouse and use the Movicon **"Properties Window"**.

Each Button can be associated three different images, which will display according to the Button's status, being button pressed, button released and button checked when the ON-OFF variable's status is not zero.

The image formats that can be used are:

- ico
- jpg
- bmp
- gif
- tif
- png
- wmf
- emf

The option to manage this image association can only be applied to certain Button types as listed below:

- Push Buttons
- 3D buttons
- Lights (Yellow, Blue, Green, Red)
- Buttons with lights (Yellow, Blue, Green, Red)
- Emergency (A,B)
- Switch
- Selector (A,B)

It is not possible to associate images to Checkboxes, Radio buttons, and Hot Regions.

When the button associated an image contains a text, the associated image to each of the three statuses will change sizes to occupy the left half of the object, leaving the right half for the text. When the button does not contain any text, the image in each of the three statuses will be resized to fit all of the button area. In this way the button can be customized by replacing, for instance, the image proposed for default by Movicon (ie. selectors and switches) with a custom image for each button status.

#### **Image Button Released**

This property is used for selecting an image to be displayed in the button when in the Released status.

#### **Image Button Pressed**

This property is used for selecting an image to be displayed in the button when in the Pressed status.

#### **Image Button Checked**

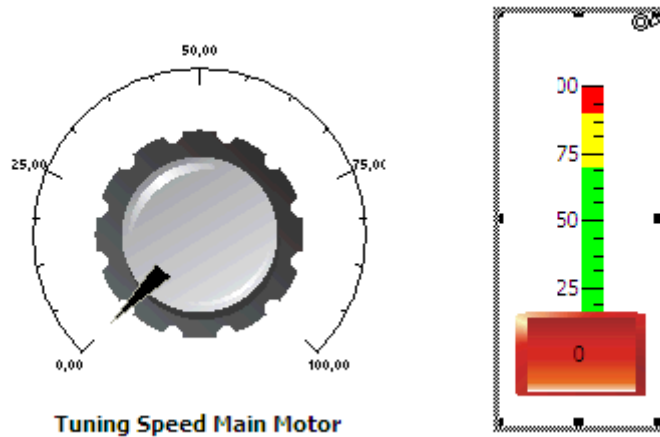
This property is used for selecting an image to be displayed in the button when in the Checked status, being when the status of its ON-OFF variable is different to zero.

# The Gauge

---

The Movicon Vectorial Gauge is a configurable object which allows specified plant variable values to be displayed or set graphically. The Gauge objects are available in the 'Slider-Gauge-Meters' group in the **"Objects Window"**.

The Vectorial Gauge is incorporated with a Slider, a Scale and a Bar which can be configured in various ways. The various components are in fact presented in the **"Objects Window"**, but they all actually derive from the Gauge's base components whose style properties have been configured in different ways.



This figure shows two Gauge examples, one circular and one vertical, each composed of one Scale, one Slider and one Bar.

## Gauge functionalities

---

The Vectorial Gauge tool has the following functionalities:

1. Setting a variable by means of a Slider
2. Displaying a variable by means of a Scale
3. Displaying a variable by means of a Filler Bar

The Gauge can be configured in different ways, for instance to get a display of one of the scale, bar or slider components only, or to display the objects concerned when required. In addition to this, different geometric shapes can be used for the Gauge display in order to simulate different objects according what is required.



# Dundas Gauges

The Dundas Gauges are a category of objects which have been implemented by Movicon by means of using external libraries. These object can be inserted from the **"Sliders-Gauges-Meters-Displays"** group in the Movicon "Toolbox". When the object is inserted on screen a wizard window will show to allow you configure the type of gauge you wish to insert with great detail:



The Dundas Gauge Configuration Window



**The Dundas Gauge configuration window is only available in English and its documentation on all the functionalities and their settings have not been included in the Movicon help. You will find the wizard, which displays when inserting Dundas gauges, is more than helpful and intuitive, allowing you to customize the object's graphical aspect in its style properties with ease.**

As you will see in the opened wizard page you can pick a gauge from any one the three object categories using the "Gauge Type Groups" list box:

- Circular Gauges
- Linear Gauges
- Numeric Indicators

After which you can go ahead and define its graphical aspect using the "Appearance Style" List-box.

After having selected the object you wish to use you can configure it by using the buttons to the right of the window (Gauges, Scales, etc.) or by using the button placed at the bottom of the window (Prev, Next, etc.). Each time you enter a configuration page you will be provided with other TAB keys at the top of the window (Position, Gauge Frames, etc.). Each object has many settings to choose letting you customize them with every little detail. When you have finished just click the "Finish" key in the Wizard window to close it and insert the object on screen. At this point you can associate the object with the variable, which is to be displayed through the "Dundas Gauge Variable" property.

All the Dundas Gauge object graphics configuration is done through the wizard window, therefore many of the properties which can be set through the Movicon Property windows will not have effect on this object. For instance, the object's "visibility" Animation property can be used but the "Back Color" Animation property will not have effect on the object and therefore can't be used.



**In the wizard window for configuring the Dundas Gauges, When opening the configuration page using the "Gauges" button and**

selecting the "Position" TAB, a "Selected Gauge" List-box appears where you can add other Gauges to the same container. By doing this more than one gauge object will appear inside the same rectangle container when inserting the object in the Movicon screen. The management of this configuration is not supported by Movicon and therefore, in this case, only the first Dundas Gauge object will be associated with the variable to be displayed.

## Editing Dundas Gauges

---

Once the Dundas gauge has been inserted on screen you can reuse the wizard to recreate the object from the beginning, or open the object's properties window to modify it. The command to use are:

- **"CTRL + click on the Open button in the object's General Properties"**: opens the wizard window and therefore restarts the object's initial configuration. Previous settings will be lost
- **"Shift + double click on the object" or "Click on the Open button in the object's General properties"**: opens the Dundas Gauge's properties window through which you can modify the already inserted object without losing any previous settings



Furthermore, you can always modify the Dundas Gauge object directly in its XML code, using the "XML Code Explorer" window for instance.



**The Dundas Gauges are more attractive to look at graphics wise compared to the normal Gauge objects, however you will need to keep in mind that more CPU will be used in those computers that don't have graphic cards with medium-high performance installed.**

## Components deriving from the Gauge object

---

The Gauge object can take on different graphical shapes and different functions according to how it is to be configured. Some of these variations are already available in the **"Sliders-Gauges-Meters"** from the **"Objects Window"**.

### Vertical Slider

The Vertical Slider is a normal Gauge changed in the following properties:

- **"Style - Type"**: refer to the gauge's graphical aspect only by changing it into a vertical shape.

### Horizontal Slider

The Horizontal Slider is a normal Gauge changed in the following properties:

- **"Style - Type"**: refers to the gauge's graphical aspect only by changing it into a vertical shape.

### Gauge Top

The Gauge Top is a normal Gauge changed in the following properties:

- **"Circular - Angle"**: changes the gauge's graphical aspect only by limiting the extension of the scale at the top part of the quadrant.

### Gauge Bottom

The Gauge Bottom is a normal Gauge changed in the following properties:

- **"Circular - Angle"**: changes the gauge's graphical aspect only by limiting the extension of the scale at the bottom part of the quadrant.

**Left Gauge**

The Left Gauge is a normal Gauge changed in the following properties:

- "Circular - Angle": changes the gauge's graphical aspect only by limiting the extension of the scale on the left hand side of the quadrant.

**Right Gauge**

The Left Gauge is a normal Gauge changed in the following properties:

- "Circular - Angle": changes the gauge's graphical aspect only by limiting the extension of the scale to the right hand side of the quadrant.

**Meter**

The Meter is a normal Gauge changed in the following properties:

- "Circular - Show Needle": changes the graphical display of the indicator represented as a needle
- "Style - Show Slider": changes the graphical display of the object's base

**Top Meter**

The Top Meter is a normal Gauge changed in the following properties:

- "Circular - Angle": changes the gauge's graphical aspect only, by limiting the extension of the scale to the top part of the quadrant
- "Circular - Show Needle": changes the needle's graphical display
- "Style - Show Slider": changes the graphical display of the object' s base

**Bottom Meter**

The Bottom Meter is a normal Gauge changed in the following properties:

- "Circular - Angle": changes the gauge' s graphical aspect only, by limiting the extension of the scales to the bottom part of the quadrant
- "Circular - Show Needle": changes the needle's graphical display
- "Style - Show Slider": changes the graphical display of the object's base

**Left Meter**

The Left Meter is a normal Gauge changed in the following properties:

- "Circular - Angle": changes the gauge' s graphical aspect only, by limiting the extension of the scales to the left hand side of the quadrant
- "Circular - Show Needle": changes the needle's graphical display
- "Style - Show Slider": changes the graphical display of the object's base

**Right Meter**

The Right Meter is a normal Gauge changed in the following properties:

- "Circular - Angle": changes the gauge' s graphical aspect only, by limiting the extension of the scales to the Right hand side of the quadrant
- "Circular - Show Needle": changes the needle's graphical display
- "Style - Show Slider": changes the graphical display of the object's base

**Vertical BarGraph**

The Vertical BarGraph is a normal Gauge changed in the following properties:

- "Style - Type": refers to the gauge's graphical aspect only by changing it into a vertical shape
- "Style - Show Slider": deletes the slider display

**Horizontal BarGraph**

The Horizontal BarGraph is a normal Gauge changed in the following properties:

- "Style - Type": refer to the gauge's graphical aspect only by changing it into a horizontal shape
- "Style - Show Slider": deletes the slider display

### **Vertical Grid BarGraph**

The Vertical Grid BarGraph is a normal Gauge changed in the following properties:

- "Style - Type": refers to the gauge's graphical aspect only by changing it into a vertical shape
- "Style - Show Slider": deletes the slider display
- "Style - Bar Brush Style": associates a non solid bar filling style

### **Horizontal Grid BarGraph**

The Horizontal Grid BarGraph is a normal Gauge changed in the following properties:

- "Style - Type": refers to the gauge's graphical aspect only by changing it into a horizontal shape
- "Style - Show Slider": deletes the slider display
- "Bar Brush Style": associates a non solid bar filling style

## **Gauge Variables Properties**

---

The Gauge Variables Properties are used for associating the Movicon Real Time DB variables to the Gauge.

To edit the Variables properties, select the object with the mouse and use the Movicon '**Properties Window**'.

### **Gauge-Slider Variable**

The name of the variable, which is to be displayed or set by different configured Gauge elements is entered in this edit box (or selected with the browse '...' button on the right).

### **Gauge-Slider Variable Min.**

The name of the variable whose value will be used as the minimum value which can be set in the '**Gauge-Slider Variable**' is entered in this edit box (or selected with the browse '...' button on the right). By doing this the threshold variable's value is made dynamic, allowing it to be also edited during the Runtime phase. In this case the Gauge Scale will be dynamically set according to the value assumed by the '**Min. Gauge-Slider Variable**'.

### **Max. Gauge-Slider Variable**

The name of the variable whose value is to be used as the maximum value that can be set in the '**Gauge-Slider Variable**' is entered in this edit box. By doing this the threshold variable's value is made dynamic, allowing it to be also edited during the Runtime phase. In this case the Gauge Scale will be dynamically set according to the value assumed by the '**Max. Gauge-Slider Variable**'.

### **Bar Offset Variable**

The name of the variable whose value is to be used as the start Bar offset.

### **MaxRate Change**

This edit box is used for entering the maximum value, in percentages, of variable changes that can be made when directly clicking on the value to be obtained instead of dragging the Gauge's needle.

### **Update value on Dragging**

When this property is enabled the variable's value will be changed while dragging the Gauge's needle. Otherwise the variable will be updated only when the Gauge's needle is released.

## Gauge Style Properties

---

The Gauge Style properties are used for setting the gauge's graphical display.  
To edit the Style properties, select the object with the mouse and use the Movicon **'Properties Window'**.

### Gauge Type

This option box is used for selecting the display mode of the Gauge's elements:

- **Vertical:** the objects enabled in the Gauge will be displayed vertically
- **Horizontal:** the objects enabled in the Gauge will be displayed horizontally
- **Circular:** the objects enabled in the Gauge will be displayed in circular mode

### Knob Style

This property allows you to choose from a default and 8 other different styles from a drop-down list which appears by clicking the arrow to the property's right. The 'default' style corresponds to a simple round knob, compared to the more complex and nice-looking 1/ 8 knob styles, and is shaped with a larger base circle with a narrower one on top for easy gripping.

### Slider Color

This property allows you to select the color to associate to the Gauge's slider.  
For further information on selecting colors please refer to the paragraph on "Color Selection".

### Knob color

This property allows you to select the color to associate to the Gauge's knob.  
For further information on selecting colors please refer to the paragraph on "Color Selection".

### Bar Back Color

This property allows you to select the color to associate to the Gauge's bar back color.  
For further information on selecting colors please refer to the paragraph on "Color Selection".

### Bar Fill Color

This property allows you to select the color to associate to the Gauge's bar color.  
For further information on selecting colors please refer to the paragraph on "Color Selection".

### Bar Brush Style

This property allows you to select the brush style to associate to the Gauge's bar.

### Scale Color

This property allows you to select the color to associate to the Gauge's scale.  
For further information on selecting colors please refer to the paragraph on "Color Selection".

### Gap

The border width of the Gauge window containing the different elements is set in this field.

### Show Slider

This enabling box is used to display or hide the Slider associated to the variable in the Gauge window. The Slider is the only Gauge element that permits the variable's value associated to the Gauge to change.

### Show Title

This enabling box is used to display or hide the Gauge object's title. The title is displayed on the top border of the window containing the Gauge.

**Show Bar**

This enabling box is used to display or hide the Filling Bar associated to the variable in the Gauge window. The Filling Bar displays the variable value associated to the Gauge in percentages.

**Show Scale**

This enabling box is used to display or hide the Scale associated to the variable in the Gauge window. The Scale is associated to the Slider to allow the current variable values of the Gauge to be displayed.

**Scale Right-Bottom**

This enabling box is used to display the Scale on the Right hand side of the Gauge window when in horizontal mode or at the bottom of the Gauge window when in vertical mode. When the Gauge is circular type the scale's text will be displayed outside the dial instead of inside.

**Bias Bar**

This enabling box is used to display or hide the Bias Bar associated to the variable in the Gauge window. This enabling only works if the Bar has already been enabled.

**3D Effects**

This enabling box is used to display the 3D effect when Gauge is set as Meter or Bargraph.

**Border**

A description of the **Border** property can be found in the paragraph on "**Style Properties common to Drawings and Controls**".

**Knob Border**

This option is used for setting the Gauge knob border's display type.

**Slider Border**

This option is used for setting the Gauge Slider's border display type.

**Bar Border**

This option is used for setting the Gauge Bar's Border display type.

**Needle Border**

This option is used for setting the Gauge Needle's border display type.

## Gauge Circular Setting Properties

---

The Setting properties for Circular Gauges are used for configuring the Gauge when set as 'Circular'. This property belongs to the Gauge's '**Style**' properties group. To edit the Circular Settings, select the object with the mouse and use the Movicon '**Properties Window**'.

**Start Angle**

The start position of the circular arc, which will represent the Gauge elements, is entered in this edit box. By editing this value you can get a longer or shorter circular arc with varying angles also in function with the value entered in the '**End Angle**' property.

**End Angle**

The end position of the circular arc, which will represent the Gauge elements, is entered in this edit box. By editing this value you can get a longer or shorter circular arc with varying angles also in function with the value entered in the '**Start Angle**' property.

**Needle Center Pos.**

This parameter allows the needle's center to be moved thus moving all the semicircle of elements, to a different position to that set for default within the Gauge window.

**Needle Width**

This selection box is used for changing the width of Gauge needle.

**Needle Length**

This selection box is used for changing the length of the Gauge needle.

**Needle Edge Color**

This property is used for selecting the color to be associated to the Gauge needle's edge.  
For further information on selecting colors please refer to "Color Selection".

**Needle Fill Color**

This property is used for selecting the color to be associated to the Gauge needle's filling.  
For further information on selecting colors please refer to "Color Selection".

**Needle Shadow Color**

This property is used for selecting the color to be associated to the Gauge needle's shadow.  
For further information on selecting colors please refer to "Color Selection".

**Show Needle**

This enabling box is used for displaying or hiding the Needle for indicating the values of the variable associated to the Gauge.

**Show Needle shadow**

This enabling box is used for displaying the or hiding the Needle's shadow for indicating the values or the variable associated to the Gauge. This enabling is only valid if the Needle's display has been enabled beforehand.

**Circular 3D**

This enables the 3D effect when the Gauge object has been configured as a Meter.

## Gauge Scale Settings Properties

---

The Gauge Scale Settings properties are used for configuring the displaying the Gauge's Scale.

This property belongs to the Gauge **'Variables'** properties group.

To edit the Scale Settings properties, select the object with the mouse and use the Movicon **'Properties Window'**.

**Min. Value**

The lowest value that the Gauge's variable may reach is entered in this edit box. This value will be also reported to the Scale's lateral borders.

**When the variable linked to the Gauge has its "Enable Scaling" property enabled, the minimum value used will be the one set in the variable's "Scale Min. Value" property.**



The gauge's **"Min. Value "** property or the variable's **"Scale Min."** property will not be considered if a variable has been entered in the Gauge's **"Min. Variable"** property for managing thresholds dynamically.

**Max. Value**

The highest value that the Gauge's variable may reach is entered in this edit box. This value will be also reported to the Scale's lateral borders.

**When the variable linked to the Gauge has its "Enable Scaling" property enabled, the minimum value used will be the one set in the variable's "Scale Max. Value" property.**



The gauge's **"Max. Value "** property or the variable's **"Scale Max."** property will not be considered if a variable has been entered in the Gauge's **"Max. Variable"** property for managing thresholds dynamically.

**# Major Div.**

The number of major divisions to be displayed in the Scale is entered in this edit box.

#### # Minor Div.

The number of minor divisions to be displayed in the Scale is entered in this edit box. The minor divisions are those comprised between two major divisions.

#### Label every # div.

For how many major divisions of the Scale a numeric label is to be displayed, identifying the value in that position, is entered in this edit box.

#### Eng. Unit

The text for identifying the engineering units of the Gauge's variable to be represented can be entered in this edit box.

#### Label Format

The type of numeric format display of the variable associated to the Gauge is set through this option box.

The formats available are represented by the following syntax:

**x**: where the x number identifies the number of figures to be displayed

**x.x**: where the x number after the decimal point indicates the number of decimal figures to be displayed

When using the decimal figure format, '**x.x**', the meaning changes according to the variable type being used whether integer or floating point. When the number is an integer the value can be displayed as a decimal (divided by 10,100, etc.). When the number is in floating point the selected decimal numbers will be displayed.



In cases where this property is not set (therefore left blank) the Gauge will inherit the format defined in its associated variable's "Default Format" property.

## Gauge Warning Zone Properties

---

The Gauge Warning Zone Properties are used for setting different background colours in the Scale according to the value assumed by the variable associated to the Gauge. By doing this one or more critical zones can be associated to the preset variable values.

This property belongs to the Gauge '**Style**' properties group.

To edit the Warning Zone properties, select the object with the mouse and use the Movicon '**Properties Window**'.

#### Enable

The Warning zone required can be activated by means of this option box. A maximum of 5 warning zones can be enabled. Enable the corresponding option boxes according to the number of zones to be created.

#### Start

A start interval value has to be entered for each enabled Warning Zone. This value is to be expressed in percentages (%) in respect to the maximum value of the variable associated to the Gauge.

#### End

An End interval value has to be entered for each enabled Warning Zone. This value is to be expressed in percentages (%) in respect to the maximum value of the variable associated to the Gauge.



Usually the '**End**' of a warning zone coincides with the '**Start**' of the next warning zone. However, if there are any overlapping values between one warning zone and the next, the value of the next warning zone will be taken as a reference.

#### Color

This property is used for selecting the colour to be associated to the Warning Zone.

For further information on the colour selection please refer to the paragraph on "Color Selection".



# Embedded Screens

---

Movicon has a special component called 'Embedded Screens' which can be inserted into Screens. This object, as implied by its name, has the job of representing Screens existing in the project and embedding them, which means inserting them inside other Screens as vectorial components.

The special feature of this object is that it contains the drawings and components contained in the associated original Screen in any scaled size desired whilst still retaining its animation features.

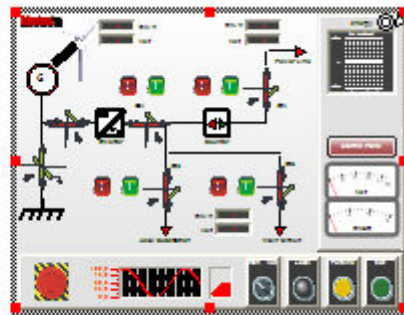
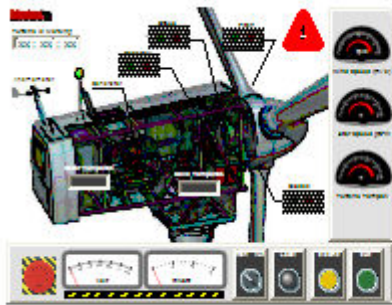
This powerful tool consents the project's general layout, for example, to be created without redesigning absolutely nothing, therefore without wasting precious time by representing ready-made Screens directly.

The object, which reproduces a Screen within another Screen can receive configuration and animation properties just like all the other vectorial components, which noticeably enhancing the potentiality and flexibility of this particular type of object.



Note: Please keep in mind that this 'Embedded Screen' object adapts to the size of the drawing in function with sizes set in the style properties of the original Screen and the container Screen.

The Embedded Screen objects are available from the 'Controls' group of the **"Objects Window"**.



## Embedded Screen Features

---

When using the Embedded Screen object you should keep in mind both its features and what is restricted in doing:

- The screen displayed by the Embedded Screen is not completely initialized. As a result the basic script interface of the screen associated to this object does not get initialized. Therefore none of the screen's VBA events can be used including variable notifications. As a consequence functions such as the "GetSynopticObject" which return the screen object containing the Embedded Screen object and not the screen displayed by the object. Events relating to the objects contained in the Embedded Screen are, however, active and managed.
- The local variables belonging to the screen displayed in the Embedded Screen are searched in the context of the screen containing the Embedded Screen.
- The IL Logic of the screen displayed by the Embedded Screen and the IL Logic of the objects it contains is executed correctly.

## Embedded Screen Style Properties

---

The Embedded Synoptic Style Properties are used for associating the Screen to be displayed.

To edit the Stile properties, select the object with the mouse and use the Movicon **'Properties Window'**.

### Embedded Screen

The name of the original Screen to be represented in the object is entered in this edit box (or selected with the '...' browse button on the far right). Therefore before this can happen the original Screen should have already been inserted in the project.

The contents of the original Screen will be tiled represented within the object and will be animated in real time with the same animations of the original Screen.



The contents of the "Embedded Screen" are refreshed only after changes made to the connected screen have been saved and the screen containing the "Embedded Screen" has been reloaded.

### Show Background

When selecting this box the background colour or any associated background drawing of the original Screen will be represented. If this box is not selected, the object will assume a transparent background.

### Static Screen Image

This property allows you to change the Embedded Screen object's behaviour as follows:

**Disabled (default):** the object will behave normally, meaning that any associated Screen animation properties will be active and the user will be able to interact within the associated Screen.

**Enabled:** the Embedded Screen will only display the image of the associated Screen and it will not be possible for the user to interact with the interactive objects internal the associated Screen. The displayed Screen can be opened by clicking on the Embedded Screen. This mode is handy for creating change pages using screen previews. It is also exploited by objects which are used in the resource for managing the "Screen Navigation Editor".



The image which is displayed when the "Static Screen Image" is active, is searched for in the same folder in which the screen resource, set in the "Embedded Screen" object, resides. The image name searched for is **"<Screen>\_c.jpg"** which is created by opening the screen in edit mode from the design environment and deleted every time the screen is modified.

### Border

The 'Border' property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

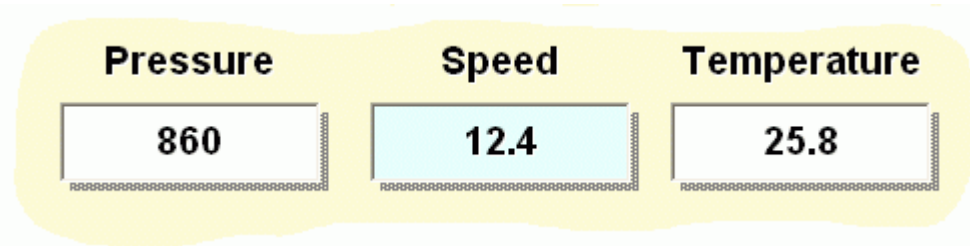
# Displays

---

The Movicon Displays are components which can be inserted into any point of the Screen to allow variables to be dynamically displayed.

The displays carry out the task of displaying figures or strings whose values are contained in the associated variable, which can be set and configured through the properties window. The Display objects are available in the 'Controls' group of the **"Objects Window"**.

The Display object also permit the operator to edit the value of the associated variable by selecting the Display and entering the value required. The Displays can also be in read only by setting the appropriate properties.



An example of Movicon Display application. The Movicon Real Time DB variables can be read and written through these Displays.

The Displays show the associated variable's value during project processing, with the style and sizes set through the **'Properties Window'**.

## Display Variables Properties

---

The Display Variables properties are used for associating the Movicon Real Time DB variables to Displays. The majority of these properties are the same ones also used in "Spin", "List" and "Combo Box" objects.

To edit the Variables properties, select the object with the mouse and use the Movicon **'Properties Window'**.

### Edit-Box Display Variable

The name of the variable to be displayed or set by using the Display is entered in this edit box (or selected with the '...' browse button on the right).

### Value to Display

This display property consents you to select the data type to display for the variable associated to the display through the "Variable Display" display property. The selectable options are:

- **Variable Value:** displays the variable's current value
- **Variable Min. Value:** displays the variable's minimum value statistic
- **Variable Max. Value:** displays the variable's maximum value statistic
- **Variable Average Value:** displays the variable's average value statistic
- **Variable Total Time ON:** displays the statistic value which indicates how long the variable stayed at a value that was not zero. In this case the value is shown in days, hours, minutes and seconds
- **Daily Variable Min. Value:** displays the variable's minimum value statistic on a daily basis
- **Daily Variable Max. Value:** displays the variable's maximum value statistic on a daily basis
- **Daily Variable Average Value:** displays the variable's average value statistic on a daily basis

- **Daily Variable Total Time ON:** displays the statistic value which indicates how long the variable stayed at a value that was not zero on a daily basis. In this case the value is shown in days, hours, minutes and seconds
- **Weekly Variable Min. Value:** displays the variable's minimum value statistic on a weekly basis
- **Weekly Variable Max. Value:** displays the variable's maximum value statistic on a weekly basis
- **Weekly Variable Average Value:** displays the variable's average value statistic on a weekly basis
- **Weekly Variable Total Time ON:** displays the statistic value which indicates how long the variable stayed at a value that was not zero on a weekly basis. In this case the value is shown in days, hours, minutes and seconds
- **Monthly Variable Min. Value:** displays the variable's minimum value statistic on a monthly basis
- **Monthly Variable Max. Value:** displays the variable's maximum value statistic on a monthly basis
- **Monthly Variable Average Value:** displays the variable's average value statistic on a monthly basis
- **Monthly Variable Total Time ON:** displays the statistic value which indicates how long the variable stayed at a value that was not zero on a monthly basis. In this case the value is shown in days, hours, minutes and seconds
- **Yearly Variable Min. Value:** displays the variable's minimum value statistic on a yearly basis
- **Yearly Variable Max. Value:** displays the variable's maximum value statistic on a yearly basis
- **Yearly Variable Average Value:** displays the variable's average value statistic on a yearly basis
- **Yearly Variable Total Time ON:** displays the statistic value which indicates how long the variable stayed at a value that was not zero on a yearly basis. In this case the value is shown in days, hours, minutes and seconds

#### **Total Time ON**

When the "Total Time ON" is displayed, the value format is shown for default as follows:

**d, HH.MM.SS**

where: d = days, H = hours, M = minutes, S = seconds

you can also customize the value to be displayed by specifying the format desired in the display's "Value Format" field. The format may include the following codes:

**%D = Days**  
**%H = Hours**  
**%M = Minutes**  
**%S = Seconds**

For example by inserting the format string:

**%H:%M:%S**

the display will now show the day value.

#### **Displaying Statistic Data**

If you wish to display the variable's statistical values you will need to enable the variable's statistics and set the variable as retentive ("Enable Statistic Data" and "Retentive not Shared" variable properties). Otherwise the variable's current value will be always displayed.

You must also take into account that the variable's statistic data will be zeroes in different ways according to the data type. For further information about this please consult the section on "Retentive Variables and Statistical Data".

#### **Default Structure**

See paragraph "Variables Properties" for Drawings and Controls.

**Min. Variable**

The name of the variable whose value will be used as the lowest value which can be set in the '**Variable**' is entered in this edit box (or selected with the '...' browse button on the right). By doing this the threshold value is made dynamic and therefore editable during Runtime.

**Max. Variable**

The name of the variable whose value will be used as the highest value which can be set in the '**Variable**' is entered in this edit box (or selected with the '...' browse button on the right). By doing this the threshold value is made dynamic and therefore editable during Runtime.

**Password Style Variable**

The name of the variable whose value is to be used for enabling the "Password" style property is entered in this edit box (or selected by using the "..." browse button on the right). This means that when this variable has a zero value, the contents in the Display Variable will be made evident. When this value is different from zero the Display Variable contents will be displayed in protected mode with asterisks and not made evident.

**Enable Var.**

See paragraph "Variables Properties" for Drawings and Controls.

**Status Var.**

See paragraph "Variables Properties" for Drawings and Controls.

## Display Style Properties

---

The Display Style properties are needed for setting the graphical display of the object. A majority of these properties are also the same ones used in the "Spin", "List" and "Combo Box" objects.

To edit the Style properties, select the object with the mouse and use the Movicon '**Properties Window**'.

**Read Only**

The display can be made read only through this selection box.

**Prompt Numeric Pad**

This option consents the Numeric or Alphanumeric Pad to display when the user clicks on the editable display. The Numeric Pad will appear when the variable is numeric type, otherwise the Alphanumeric Pad will appear. The Pad is always opened in the center of the screen, and shows the variable's current values and the minimum and maximum limits as set in for the display object. In addition to this, these values are formatted using the format set in the display object.



This option may be enabled automatically by Movicon when if the system being used has a Touch-Screen. Therefore, to determine this, check to see whether the system has not got a mouse or has one with just one button.

**Has Spin**

By using this check box you can associate Spin button to the display so that the variable's value can be increased or decreased without having to use the keyboard.

**Password**

This property, when enabled, is used for protecting the display's contents. Each character will be displayed as an asterisk during and after editing. This is used when editing texts, such as passwords, whose original characters must not be evident.

**Horizontal Spin**

By using this check box you can set the Spin to be horizontal or vertical.

### Spin Delay

The delay time in enabling the fast Increase/Decrease function is entered in this edit box. By pressing one of the two spin buttons with the mouse for the time set, when exceeded the variable will start to Increase/Decrease in fast mode.

### Spin Size

When the Spin button associated to the display has been enabled you can select the button's size from the ones below:

- small
- medium
- large

### Min. Value

This edit box is used for specifying the minimum value which the Display variable can obtain when this display is being written in.

**When the variable linked to the display has its "Enable Scaling" property enabled, the minimum value used will be the one set in the variable's "Scale Min. Value" property.**



The display's "Minimum Value" property or the variable's "Scale Min." property will not be considered if a variable has been specified in the display's "Min. Variable" property for managing the threshold dynamically.

### Max. Value

This edit box is used for specifying the maximum value which the Display variable can obtain when this display is being written in. When the associated variable is string type this value will be considered as the maximum number of characters which can be inserted in the variable.

**When the variable linked to the display has its "Enable Scaling" enabled, the maximum value used will be the one set in the variable's "Scale Max. Value" property.**



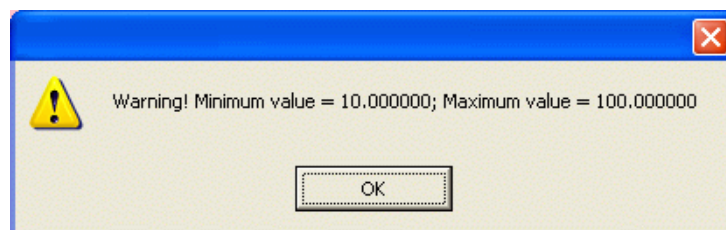
The display's "Maximum Value" property or the variable's "Scale Max." property will not be considered if a variable has been specified in the display's "Max. Variable" property for managing the threshold dynamically.

### Error String

This edit box can be used for inserting a text string which will displayed in a MsgBox each time an attempt is made to set a value not within the preset limits. The minimum and maximum limits allowed can also be displayed by using the "%f" syntax. An example string is shown below:

Warning! Minimum value = %f; Maximum value = %f

In runtime this would result as follows:



### Spin Step

In this edit box the increment or decrement value can be entered to be applied to the variable by using the spin buttons.

### Eng. Unit

A text to identify the Display variable's engineering unit to be represented is entered in this edit box.

### Format

The associated Display variable's numeric format to be displayed is selected through this box.

The type of formats available are represented by the syntax below:

**x**: where the x number identifies the number of figures to be displayed

**x.x**: where the x number after the decimal point indicates the number of decimal figures to be displayed

When using the 'x.x' decimal figure format its meaning changes according to variable type whether being 'integer' or 'floating point'. When being an 'integer' number, the value can be displayed as a decimal (divided by 10,100,etc.). When being a 'floating point' number, the selected decimal numbers will be displayed.

#### Example 1

If you want to display a value formatted with three figures, select the '**xxx**' Format. The result will be according to how the variable's value is:

Variable Value	Value Displayed
1	001
10	010
100	100

#### Example 2

If you want to display a value formatted with two decimal figures, select the '**x.xx**' Format. The result will be according to how the variable's value is:

Variable Integer Value	Value Displayed
1	0.01
10	0.10
100	1.00

Variable Float Value	Value Displayed
1.2345	1.23



When using formats with decimal figures applied to integer Variables, only their display and not their contents can be edited. In practice divisions of 10, 100, etc., is only for graphical displaying purposes and does not influence the real value of the variable in any way.



**When using basic expression in symbols Text Animation Property, it must be set the format "%f".**

When integer type variables link to the display with a format for showing decimals, the minimum, maximum limits and the spin step do not keep to the format set in the display and will always work on the variable's total value. For example, for a "x.x" format and a maximum limit of "100", it will be possible to edit a maximum value equal to "100" in the variable to be shown on display with a "10.0" maximum.



In cases where this property is not set (therefore left blank) the display will inherit the format defined in the associated variable's "Default Format" property.

#### Border

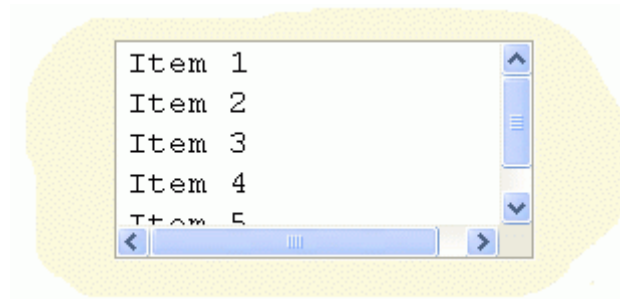
The '**Border**' is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

# The List Objects

---

The Movicon List objects are components which can be inserted in at any point of the screen so that item lists can be displayed. A numeric or string type variable can be associated to the object. Each line of the list corresponds to a numeric value which is made available on the variable associated to the object when this variable is numeric type. When the variable is string type, this will be updated with the selected Item's text. Each time a line is selected the list's variable is updated with the corresponding numeric value (the value is progressive, starting from 0 to indicate the first line) or with the selected Item's text.

The List objects are available from the 'Special Objects' group in the **"Object Window"**.



An example of a Movicon List.

The list will display the Items during project processing with the style and sizes setup through the **'Properties Window'**.

The List object can be managed and compiled by using the component's Basic Script functions (ListBoxCmdTarget interface).

## ListBox Execution Properties

---

The ListBox Execution properties allow you to associate the Variable and Items to be displayed.

To change the Execution properties, select the object the mouse and use the Movicon **'Properties Window'**.

### Variable ListBox

The name of the variable in which the numeric value or text corresponding to the selected Item is entered in this box (or selected with the "..." browse button on the right). The variable can be either numeric or string type. When the variable is numeric the selected item's index will be written in it, starting from the "0" value. When the variable is string type the text corresponding to the selected item will be written in it.

### Items ListBox

The texts which are to appear in the list are entered in this box. To insert more than one Item you need to separate the texts with the pipe character "|" (i.e. Item1|Item2|Item3|etc.). As well as fixed texts you can also insert String IDs which are presented in the **"String Table"**. In this case you only need to create a String ID and insert all the Item's texts in its internal which are to appear on the list using the 'pipe' character ("|").



You can compile the List dynamically by using the object's appropriate Basic functions.



## List Style Properties

---

The List's Style properties are used for setting the object's graphical settings. To change the Style properties, select the object with the mouse and use the Movicon **"Properties Window"**.

### Border

The **"Border"** property is described in the paragraph headed **"Style Properties common to Drawings and Controls"**.

### Clickable

The **"Clickable"** property is described in the paragraph headed **"Style Properties common to Drawings and Controls"**.

### Show Control Wnd

The **"Show Control Wnd"** property is described in the paragraph headed **"Style Properties common to Drawings and Controls"**.

### ListBox Items

The texts to appear in the list are inserted in this edit box. To insert more than one item you need to use the "|" pipe character to separate them (ie. Item1|Item2|Item3|etc.). ID strings, existing in the **"String Table"**, can also be inserted as well as the Item texts and also need to be separated with the "pipe" ("|") character.



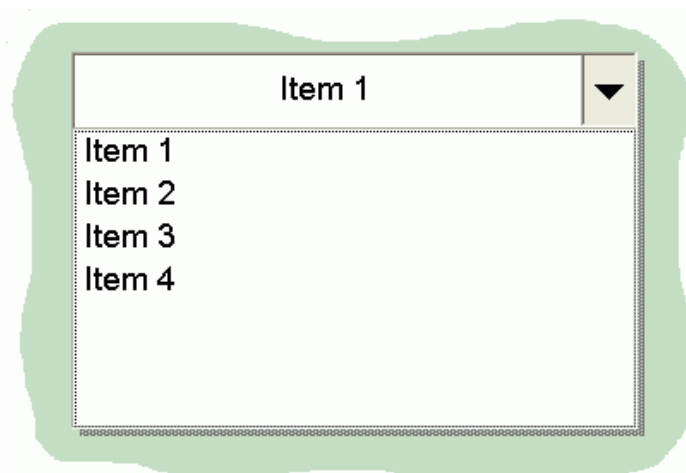
The list can be dynamically compiled by using the objects appropriate Basic functions.

## Combo Box Objects

---

The Movicon Combo Box objects are components which can be inserted into any point to the screen to allow item lists (texts) to be displayed and edited. The object can be associated with a numeric or string type variable. Each line of the list corresponds to a numeric value which is made available in the variable associated to the object, when this variable is numeric type. When the variable is string type, it will be updated with the text of the Item selected. Each time a line is selected the Combo Box's variable is updated with the corresponding numeric value (the value is progressive, starting from 0 to indicate the first line) or with the text of the Item selected. However, when the variable is numeric type, no values are displayed in the Combo Box's editing box, therefore a string variable type is normally used. In addition to the Combo Box a second string type variable can be associated which will be compiled with the text of the item to be displayed.

The Combo Box objects are available from the "Special Object" group in the **"Object Window"**.



Movicon Combo Box example

The Combo Box will display the Items during project processing, with the style and sizes setup through the **"Properties Window"**.  
You can also insert new items with the Combo Box by using the appropriate edit boxes.

*The managing and compilation of Combo Box objects can also be done through the component's Basic Script functions (DisplyCmdTarget and ListBoxCmdTarget interface).*

## Combo Box Style Properties

---

The Combo Box Style properties are used for setting the graphical properties of the object. To change the Combo Box properties, select the object with the mouse and use the Movicon **"Properties Window"**.

The Combo Box is a combination of a **"Display"** and a **"ListBox"**. All its Style properties are therefore the same described for the two Display and ListBox components.

### Read Only

The **"Ready Only"** property is described in the paragraph headed **"Display Style Properties"**.

### Spin Size

The **"Spin Size"** property is described in the paragraph headed **"Display Style Properties"**.

### Min. Value

The **"Min. Value"** property is described in the paragraph headed **"Display Style Properties"**.

### Max. Value

The **"Max. Value"** property is described in the paragraph headed **"Display Style Properties"**.

### Eng. Unit

The **"Eng. Unit"** property is described in the paragraph headed **"Display Style Properties"**.

### Format

The **"Format"** property is described in the paragraph headed **"Display Style Properties"**.

### Show Control Wnd

The **"Show Control Wnd"** property is described in the paragraph headed **"Style properties common to Draws and Controls"**.

### ListBox Items

The **"ListBox Items"** property is described in the paragraph headed **"Display Style Properties"**.

### Border

The **"Border"** property is described in the paragraph headed **"Style properties common to Draws and Controls"**.

## Combo Box Variables Properties

---

The Variable properties of Combo Box allow associations of the Variables and the Items to be displayed.

To change the Variables properties, select the object and use the Movicon **"Properties Window"**.

The Combo Box is a combination of a **"Display"** and a **"ListBox"**. All its Variables properties are therefore the same as described for the two Display and ListBox components.

#### **EditBox-Display Variable**

This property is described in the paragraph headed **"Display Variables Properties"**. You must also consider that when this variable is string type it will contain the text of the selected item and when it is numeric type it will contain the index number of the selected item. In this case, however, the text selected will be shown in the display while the index will be returned to the variable. The item's index starts from the "0" value.

#### **ListBox List Variable**

This property is described in the paragraph headed **"ListBox Execution Properties"**.

#### **EditBox-Display Min. Variable**

This property is described in the paragraph headed **"Display Variables Properties"**.

#### **EditBox-Display Max. Variable**

This property is described in the paragraph headed **"Display Variables Properties"**.

#### **Tag Stile Password**

The **"Tag Stile Password"** property has been described in the "Display Variables Properties" paragraph.

## TAB Group Object

---

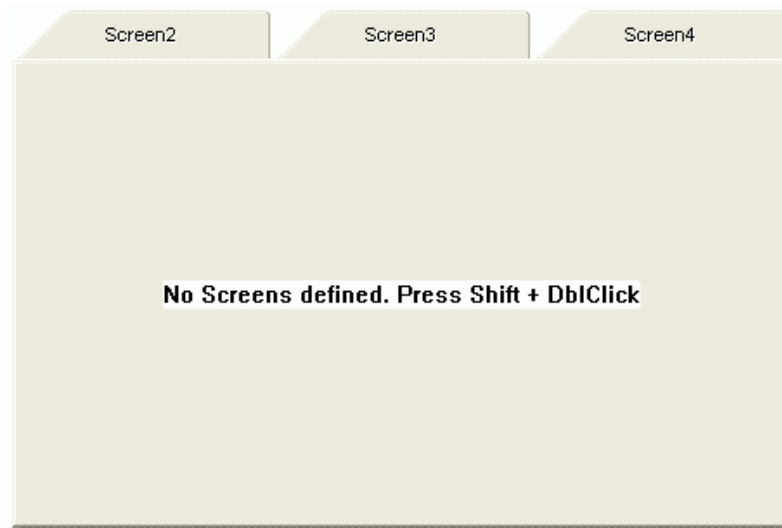
The Movicon TAB Group objects are components which can be inserted into any point of the screen. This object's task is to display the project's Screens within its window. This is done by selecting the desired screen's corresponding TAB. The list of screens available should be put together in the object's programming phase.

The particularity of these objects is that they represent, in scale, the symbols and components which are contained in the Screens they are associated to while conserving any animation characteristics.

This powerful feature allows you to realize, for example, the general layout of the project, without having to redraw absolutely anything, by using the ready-to-use Screens.

This particular object, which reproduces a screen inside another screen, can receive its own configuration and animation properties just like all the other vectorial components, which remarkably enhance its potentiality and flexibility.

The TAB Group objects are available from the "Special Objects" group found in the **"Objects Window"**.





The tags of the "TAB Group" object adapt to the length and height of the text starting from the top when there isn't enough space horizontally.

The object's tabs are headed with the names of the associated screens. To custom these texts you will need to insert a String ID in the String Table with the name of the screen to be displayed. In this way the text will automatically be replaced with the string contents. When screens are in resource groups you will need to specify the group name as well:

ID = Group1\Synoptic1

## TAB Group Object Features and Constraints

---

When using the TAB Group object you will need to keep in mind some of its characteristics and restrictions of use:

- The TAB Group object keeps all loaded screens within its internal memory and therefore all the variables contained in these screens and exchanged with the field will always remain in use independently from the screen displayed by the object. This function serves for quick screen navigation and no option exists for Movicon to discharge memory when passing from one screen to another
- The screens displayed by the TAB Group object are not completely initialized. As a result the screen basic interfaces do not initialize. Therefore, none of the screen VBA events can be used, including variable notifications. As a consequence, functions such as the "GetSynopticObject" return the screen object which contains the TAB Group object and not the screen displayed by the object. Events relating to objects contained in Embedded Screens are however active and managed.
- Screen Local variables displayed by the TAB Group are created in the context of the Screen that contains the TAB Group object.
- Screen IL logic displayed by the TAB Group and IL logic of the objects it contains is executed correctly.

## TAB Group Style Properties

---

The TAB Group Style properties are used for setting the object's graphic aspect. To edit the Style properties, select the object with the mouse and use the Movicon "**Property Window**".

### **Flat**

This property displays the TAB flat and not raised.

### **Wider 3D Style**

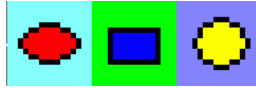
This property displays the TAB in 3D. This setting has only effect when the "**Flat**" property is disabled.

### **Bottom TabBar**

This property displays the TAB bar at the bottom of the object instead of at the top.

### **Tab Image**

This property allows you to select an image to be displayed at the side the text on the label of each tab. The associated image must be sized 16x16 pixels for each screen set in the "Tab group" object. The complete image to be associated to the property should measure 16 pixels in height with length equal to the of screens multiplied by 16. For example, if you insert three screens in the TAB Group, the associated image will look like this:



the completed size is therefore 16x48 pixels. The end result will look like this on the TAB Group object:



## TAB Group Execution Properties

---

The TAB Group Execution properties are used for setting up the list of screens to be displayed in the object.

To change the Execution properties, select the object with the mouse and use the Movicon **"Property Window"**.

### Edit Screen List

When executing this command a dialog window will open through which you can select the screen list you wish to associate to the object.

The current screen list is shown on the left hand side of the window. Other screens can be added to the list by using the **"Add New Screen...."** button while the **"Remove"** button is used to remove the selected screen from the list. To change the order of list just click on the screen to be moved and drag it to the position desired.

# Alarm Windows

The Movicon Alarm Window is a configurable object designed to represent and control local or area plant alarms, or remote station alarms (Server) connected in net.  
The Alarm Window is available from the 'Controls' group in the **"Objects Window"**.

Alarm Text	Alarm On	Alarm Off	Duration	Priority	Stat..	Ack All (Ctrl+A)
Temperature TK1 too high	31/03/2008 17:20:49	31/03/2008 17:53...	00:30:25	1	OFF	
ALARM ON	2008-04-12 10:43:39					
Laser Door Opened	02/04/2008 15:03:02	10/04/2008 15:00...	23:57:23	1	OFF	
Turbine Temperature Alarm	12/04/2008 9:22:09			1	ACK	
Engine Temp. 2 Alarm	27/03/2008 10:46:49	27/03/2008 11:11...	00:46:47	1	OFF	
Sensor Alarm Machine 2 - Ongrt	10/04/2008 17:00:25	12/04/2008 9:22:09	16:21:54	1	OFF	
Sensor Alarm Machine 3 - Ongrt	02/04/2008 13:59:47	02/04/2008 13:59...	00:00:01	1	OFF	
Sensor Alarm Machine 4 - Ongrt	11/03/2008 16:54:58	11/03/2008 16:55...	00:00:01	1	OFF	
Sensor Alarm Machine 1 - Ongrt	10/04/2008 15:21:20	10/04/2008 15:21...	00:00:01	1	OFF	

When the Alarm Window is inserted into the Screen it will appear with default settings.  
After having inserted the Alarm Window it can be sized as pleased by dragging its borders with the mouse.



A number of Alarm Windows can be inserted into the project, ie. in more Screens, as required.

In the Alarm Window alarms supporting Reset and Ack have to be acknowledged and reset to disappear from the window, and alarms not supporting the Reset and Ack, and therefore result as simple Messages, appear and disappear automatically in function with the ON/OFF status of the alarm itself.

## Alarm Window Fields

The fields to be presented in the Alarm Window can be selected by the programmer from those available through the **"Field Choice"** window.  
The fields or columns that can be displayed are as follows:

### Alarm Description

The Alarm Description column reports descriptive text of the alarm. The text is composed of strings in the following formations:

Text	Condition
Device Name - Title	This formation will be displayed when a text has been inserted in either the 'Alarms->General->Device Name' property or the 'Threshold->General->Title' property.
Device Name - Name	This formation will be displayed when a text has been inserted in the 'Alarms->General->Device Name' property but not in the 'Threshold->General->Title' property. In this case the text of the 'Threshold->General->Name' will be considered.

<b>Title</b>	This formation will be displayed when a text in the 'Alarms->General->Device Name' property has not been inserted but only in the 'Threshold->General->Title' property.
<b>Name</b>	This formation will be displayed when a text has not be entered in either the "Alarms->General->Device Name" or the "Threshold->General->Title" properties. In this case the 'Threshold->General->Name' property will be considered.
<b>Variable Name - Title/Caption</b>	This formation will be displayed when the template alarm associated to a variable has been used. Therefore the name of the variable and the text inserted in the "Threshold->General->Title" will be considered.
<b>Variable Name - Name</b>	This formation will be displayed when the template alarm associated to the variable has been used. The name of the variable and the text inserted in the Threshold->General->Name' property will be considered if a text has not been inserted in the "Threshold->General->Title" property.

#### Time ON

The 'Time ON' column reports the date and time of alarm occurrence. The time will always be that of operating system's and can be displayed with the formation specified in the **"Time Format"** property from the **'Alarm Window Style Properties'**. Any Clients connected to the Server will display the time in their local format.

#### Time ACK

The 'Time ACK' column reports the date and time of the alarm's acknowledgement. The time will always be that of the operating system's local time and can be displayed with the formation specified in the **"Time Format"** property from the **'Alarm Window Style Properties'**. Any Clients connected to the Server will display the time in their local format.

#### Time OFF

The 'Time OFF' column reports the date and time of the alarm's deactivation. The time will always be that of the operating system's local time and can be displayed with the formation specified in the **"Time Format"** property from the **'Alarm Window Style Properties'**. Any Clients connected to the Server will display the time in their local format.

#### Time RESET

The 'Time RESET' column reports the date and time of the alarm's reset. The time will always be that of the operating system's local time and can be displayed with the formation specified in the **"Time Format"** property from the **'Alarm Window Style Properties'**. Any Clients connected to the Server will display the time in their local format.

#### Total Time ON

the "Total Time ON" column displays the total time in which the alarm remained active from the moment in which it was initialized. The "Total Time ON" value is set in hours, minutes and seconds for default:

**H:M:S**

where: H = hour, M = minutes, S = seconds

You can however customize how the value is to be displayed by specifying the desired format in the Alarms Window "Duration Format" property. The formats may include the following codes:

%D = Days  
%H = Hours  
%M = Minutes  
%S = Seconds

For instance, when inserting the format string:

%D.%H:%M:%S

the total duration value will also include the number of days (ie. "5.12:23:45").

### Image

The "Image" column shows the image which has been associated to the variable in the **"Variable Options Properties - BMP File"**, which created the alarm.

### Status

The 'Status' column reports the current status of the alarms. The Alarm status are:

- **ON:** alarm active and not acknowledged
- **OFF:** alarm not active and not acknowledged
- **ACK:** alarm active and acknowledged
- **OFF (ACK):** alarm not active and acknowledged

### Duration

The 'Duration' Column reports the duration of the alarm occurrence. The text with a customized time format as described in the **"Duration Format"** property from the **"Alarm Window Style Properties"** can also be inserted.

### Severity

The 'Severity' column reports the alarm priorities. This value should have been previously inserted in the **"Severity"** property from the **"Alarm Threshold Execution Properties"**.

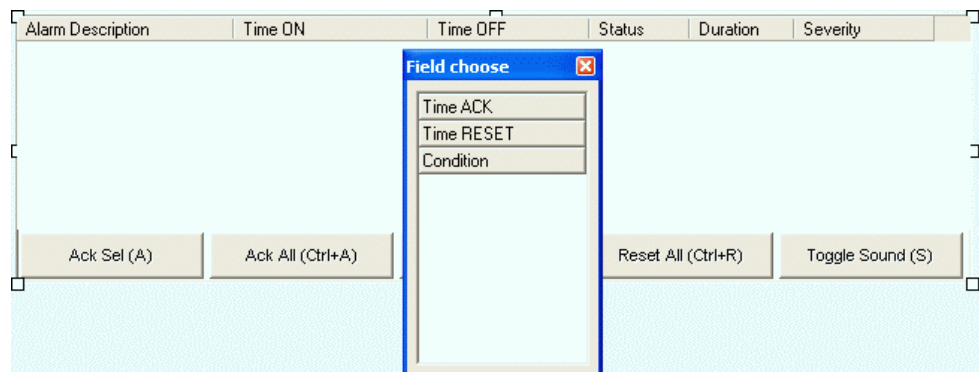
### Condition

The 'Condition' column reports the alarm's current condition. The options are:

- **ON:** alarm active
- **OFF:** alarm not active

## Alarm Window Field Choice

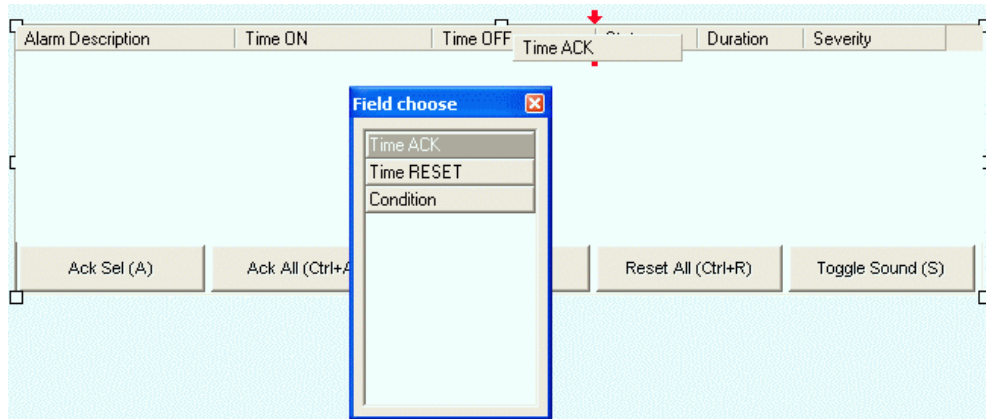
The Alarm Window allows you to select the fields or columns which can be displayed. In order to carry out this operation you need to open the **'Field Choice'** window by clicking on the **'Open'** button in the **'General'** group of the object's **'Properties Window'**, or by double-clicking on the object and pressing down the **'SHIFT'** key at the same time.



In the 'Field Choice' window are listed available fields which have not yet been inserted into the Alarm Window. To move a field from 'Field Choice' window to the Alarm Window



simply select it with the mouse and drag it to the position desired in the Alarm Window. Please bare in mind that the field can only be released at the beginning or the end of a already positioned field, and two red arrows should appear:



To move a field from the Alarm Window to the 'Field Choice' window, simply select it with the mouse and drag it to the 'Field Choice' window.



When the window is set with the "Show Control Wnd" option during Runtime phase the columns can then be dragged out of the window and deleted. However the initial configuration must be restored by using the "RecalcLayout" basic script method from the AlarmWndCmdTarget interface or when the page is reloaded.

## Alarm Help in the Alarm Window

A string help can be associated to each single alarm through the **"Help"** property found in the 'Alarm Threshold General Properties'. This help string can be displayed during Runtime in the Alarm Window by double-clicking on the line of the alarm required.

When alarms are managed as Templates and associated to variables without the associated help strings, the description of the variables will be displayed.

In addition to this you can also get more extended help associated to each single alarm by using a HTML or CHM file. To open these files, just select the alarm in the Alarm Window and click the **"Help"** key. Obviously, this can only be done when the help files have already been created and configured in one of the two ways:

1. Associate a "htm" file to the variable which generates the alarm. This setting must e done in the **"Html File"** property from the "Variable Options Properties" group. In order to avoid absolute file path problems if would be advisable to insert the "htm" files into the project's resource folder.
2. Associate a help "chm" file format to the project, and create a topic referring to the alarm/s. Then insert the topics in the chm file's index and give them the same name of the varialbe associated to the alarm. For instance, if the "Alarm001" has been set with the "Alarm\_1" variable associated to it, the topic will be inserted into the file index in the name of "Allarme\_1"

## Alarm Window Style Properties

The Alarm Window Style Properties are used for setting the object's graphical properties. To edit the Style properties, select the object with the mouse and use the Movicon **'Properties Window'**.

**Border**

The '**Border**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

**Clickable**

The '**Clickable**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

**Auto Layout**

The '**Auto Layout**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

**Show Control Wnd**

The '**Show Control Wnd**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

**Show Preview**

This property allows you to activate the help to show for default for each alarm row. Otherwise the help will only show when the alarm row is double clicked on. This only works for those alarms which have a help string.

**Ack Sel Button**

This option enables the command button for acknowledging the alarms selected in the Alarm Window. One or more alarms have to be selected in order to carry out this action in Runtime.



This button is made active only when an alarm is selected. When no alarm has been selected, this button will result disabled.

**Ack All Button**

This option enables the command button for acknowledging all the alarms existing in the Alarm Window, without having to select them all first.

**Reset Sel Button**

This option enables the command button for resetting the alarms selected in the Alarm Window. One or more alarms, previously acknowledged with the 'Ack sel' or 'Ack All' button, have to be selected in order for this to work in Runtime.



This button is made active only when an alarm is selected. When no alarm has been selected, this button will result disabled.

**Reset All Button**

This option enables the command button for resetting all the alarms existing in the Alarm Window, without having to select them first and only if previously acknowledged with the 'Ack Sel' or 'Ack All' button.

**Toggle Sound Button**

This option enables the command button for activating or deactivating the acoustic signal associated to the priorities of unacknowledged alarms. This button is latched. Therefore, it is possible to tell if the sound is active or not based on the button's graphical state (pressed or released). The button's status is also retentive, therefore its settings are retained even when the project is closed and reopened. The button's status is directly linked to the corresponding system "AlarmSoundState" variable's status. Modifying the status of one will automatically modify the other and viceversa.



**The Sound Button is not enabled in Client (or WebClient) project Alarm Windows, but only in Server projects. The sound of alarms is managed by the Alarm Window only for local project alarms and not for those alarms retrieved from Network Server projects.**

**Help Button**

This selection provides the command button for opening the help file of the alarm selected. The alarm's help file is executed as explained in the section on "**Alarm Help in the Alarms Window**".



This button is made active only when an alarm is selected. When no alarm is selected this button results disabled.



**The Help Button is not enabled in Client (or WebClient) project Alarm Windows, but on in Server projects. The alarm Help is managed by the Alarm Window only for local project alarms and not for those alarms retrieved from Network Server projects.**

#### **Get History Button**

This selection makes the command button available for displaying the selected alarm's history, being all of the "Alarms Historical", showing all of the status transactions which verified while the alarm was working.



This button is only activated when an alarm is selected. This button will remain disabled when no alarm has been selected.

The alarm's history can be retrieved by using this button or by pressing "Shift + Doppio Click" on the alarm or by selecting the alarm and pressing the "G" key. This functionality is also supported by the Alarm Window when connected to a Network Server project.

#### **Button Size**

The '**Button Size**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

#### **Align Buttons**

The '**Align Buttons**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

#### **Ack Sel Button Text**

The text which is to appear on the '**Ack Sel Button**' is entered in this edit box. The default text will be used if left blank.

#### **Ack All Button Text**

The text which is to appear on the '**Ack All Button**' is entered in this edit box. The default text will be used if left blank.

#### **Reset Sel Button Text**

The text which is to appear on the '**Reset Sel Button**' is entered in this edit box. The default text will be used if left blank.

#### **Reset All Button Text**

The text which is to appear on the '**Reset All Button**' is entered in this edit box. The default text will be used if left blank.

#### **Toggle Sound Button Text**

The text which is to appear on the '**Toggle Sound Button**' is entered in this edit box. The default text will be used if left blank.

#### **Help Button Text**

The text to appear on the "**Help Button**" is inserted in this edit box. The default text will be used if this field is left blank.

#### **Get History Button Text**

The text to appear in the "**Get History Button**" is inserted in this edit box. The default text will be used if this field is left blank.

#### **Text Column Name**

The text which is to appear as the "**Alarm Description**" column's name is entered in this edit box. The default text will be used if left blank.

#### **Alarm On Column Name**

The text which is to appear as the "**Time ON**" column's name is entered in this edit box. The default text will be used if left blank.

#### **Alarm Ack Column Name**

The text which is to appear as the "**Time ACK**" column's name is entered in this edit box. The default text will be used if left blank.

**Alarm Off Column Name**

The text which is to appear as the "**Time OFF**" column's name is entered in this edit box. The default text will be used if left blank.

**Alarm Reset Column Name**

The text which is to appear as the "**Time RESET**" column's name is entered in this edit box. The default text will be used if left blank.

**Alarm Duration Column Name**

The text which is to appear as the "**Duration**" column's name is entered in this edit box. The default text will be used if left blank.

**Alarm Total Time ON Column Name**

The text which is to appear as the "**Total Time ON**" column's name is entered in this edit box. The default text will be used if left blank.

**Alarm Severity Column Name**

The text which is to appear as the "**Severity**" column's name is entered in this edit box. The default text will be used if left blank.

**Alarm Status Column Name**

The text which is to appear as the "**Status**" column's name is entered in this edit box. The default text will be used if left blank.

**Alarm Condition Column Name**

The text which is to appear as the "**Condition**" column's name is entered in this edit box. The default text will be used if left blank.

**Alarm Image Column Name**

The text to appear as the "**Image**" column's name is inserted in this edit box. The default text will be used if left blank.

**Time Format**

The date and time format which is to be displayed in the 'Time On', 'Time Ack', 'Time Off' and 'Time Reset' columns is entered in this property. All the format codes which can be used in this property are listed below:

Code	Meaning
%a	Abbreviated name of Weekday
%A	Complete name of Weekday
%b	Abbreviated name of month
%B	Complete name of month
%c	Local Time and Date
%d	Day of month expressed in numbers (01-31)
%H	Time expressed in 24 hours ( 00-23)
%I	Time expressed in 12 hours (01-12)
%j	Day of year expressed in numbers (001-366)
%m	Month expressed in numbers (01-12)
%M	Minutes expressed in numbers (00-59)

<b>%p</b>	A.M./P.M. indicator for 12 hr representation
<b>%S</b>	Seconds expressed in numbers (00-59)
<b>%U</b>	Weeks of the year expressed in numbers, with Sunday as first day of the week (00-53)
<b>%w</b>	Days of the Week expressed in numbers (0-6; Sunday = 0)
<b>%W</b>	Weeks of the year expressed in numbers, with Monday as first day of the week (00-53)
<b>%x</b>	Local date
<b>%X</b>	Local time
<b>%y</b>	Year expressed in numbers without specifying century (00-99)
<b>%Y</b>	Year with century expressed in numbers.
<b>%z,%Z</b>	Time difference abbreviated or in full
<b>%%</b>	Percentage sign

The '#' character can be used for further modifying some formats to get the following results:

Code	Meaning
<b>##a, ##A, ##b, ##B, ##p, ##X, ##z, ##Z, ##%</b>	The "#" character has no meaning to these formats and therefore will be ignored
<b>##c</b>	Local date and time in full format. Example: "Friday, August 06, 2004, 08:48:55"
<b>##x</b>	Local time represented in full format. Example: "Friday, August 06, 2004"
<b>##d, ##H, ##l, ##j, ##m, ##M, ##S, ##U, ##w, ##W, ##y, ##Y</b>	The "#" character in these formats deletes any zeros preceding figures. Example: "05" becomes "5"

#### Duration Format

The date and time format to display the times in the Alarms Window's "Duration" and Total Time ON" columns is entered in this property. The message will be updated only on 'Alarm Off' events.

The message may include the following special codes only:

- **%D** = Days
- **%H** = Alarm duration in hours
- **%M** = Alarm duration in minutes
- **%S** = Alarm duration in seconds

When this field is left empty, Movicon will automatically insert the alarm's total duration in the following formation:

00:00:00

where 00:00:00 indicates the time of the alarm's duration in hours, minutes and seconds.

#### Include milliseconds

Milliseconds will be included in the 'Time' formats of the 'Time On', 'Time Ack', 'Time Off' and 'Time Reset' columns when this check-box is enabled.

#### Area Filter

A filter can be inserted in this property to display the alarms belonging to a certain area only. The filter can contain one or more special characters "\*" (ie. \*Area\*).

#### Alarm Text Filter

A filter can be inserted in this property to display alarms based on their text. The filter is applied to the 'Alarm Description' column and can contain one or more special characters "\*" (ie. \*Turbine\*).

#### Alarm Severity Filter

A filter can be entered in this box for displaying the alarms based on their priorities. The value inserted must be a severity value (value "1" indicates the lowest severity) you wish to use for executing the filter. Once the severity value has been set you will need to also set the "Alarm Severity Filter Condition" with the condition type with which to apply the filter. Leaving this value at zero (default value), no filters will be applied based on the alarms' severities.



The filter by severity is also supported in cases where the "alarms Window" is used for retrieving alarms from a Network Server Project (therefore both in Networking and WebClient).

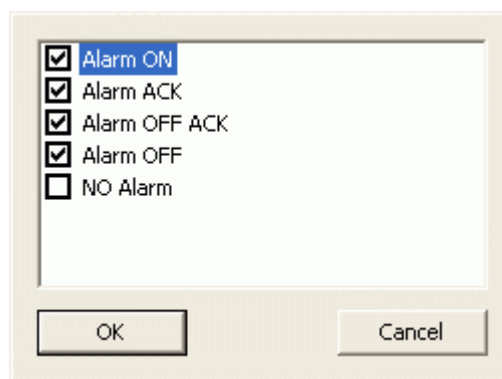
#### Alarm Severity Filter Condition

This property consents you to set the condition to be used by the alarm filter based on severity. When the "Alarm Severity Filter" is left at "0" value, this setting will be ignored. The options are:

- **equal:** alarms with severity equal to the one set in the "Alarm Severity Filter" property will be displayed only
- **minor:** alarms with lower severity to the one set in the "Alarm Severity Filter" property will be displayed only
- **major:** alarms with higher severity to the one set in the "Alarm Severity Filter" property will be displayed only

#### Alarm Mask Filter

A filter based on the status of the alarms to be displayed in the window can be set in this property. Click on the '...' button on the far right of the property to open the following settings window:



The filters are:

- **Alarm ON:** active alarms which have not yet been acknowledged will be displayed

- **Alarm ACK:** active alarms which have been acknowledged but not yet reset will be displayed
- **Alarm OFF ACK:** inactive alarms which have been acknowledged but not reset will be displayed
- **Alarm OFF:** inactive alarms which have been either acknowledged or not acknowledged but not yet reset will be displayed
- **NO Alarm:** inactive alarms which have already been acknowledged and reset will be displayed

#### **Historical Log Item Back Color**

This property allows you to select the color to associate to the background of the alarm historical data display area. To view alarm historicals use the "Get History" button. Colors are selected using the standard selection procedures using the color palette.

#### **Historical Log Item Text Color**

This property allows you to select the color to associate to the text in the alarm historical data display area. To view alarm historicals use the "Get History" button. Colors are selected using the standard selection procedures using the color palette.

## **Alarm Window Execution Properties**

---

The Alarm Window Execution Properties are used for setting the eventual Server names from where data to be viewed is taken.

To edit the Execution properties, select the object with the mouse and use the Movicon '**Properties Window**'.

#### **Blink Time**

This setting represents the blink time for alarms not yet acknowledged. The value is expressed in milliseconds.

#### **Network Server**

See paragraph "Drawings and Controls common Execution Properties".

#### **Network Backup Server**

See paragraph "Drawings and Controls common Execution Properties".

#### **Network Polling Time**

This setting represents the network's polling time. The value is expressed in milliseconds.

#### **OPC AE Server**

The name of the eventual OPC AE Server, from which alarm notifications are to arrive and be displayed in the window, is entered in this edit box. In this way the alarm window will be configured so that it keeps track of a certain number of events in arrival from a OPC Server AE. In addition to this, the "OnOPCAEEvent" is available from the "AlarmWndCmdTarget" basic interface which consents you to intercept a new AE type message in arrival for the alarms window, and to eventually stop it from being displayed.

#### **Max. OPC AE Events Nr.**

The max. number value of OPC Server AE events that can be displayed in the Alarms window is entered in this field. When this set limit has been reached the oldest traces will be replaced with the newest ones.

# Log Window

---

The Movicon **'Log Window'** is a configurable object designed to display log data recorded by the Project. Data from local projects and remote stations (Server) connected in net can also be displayed.

The Log Window object is available from the 'Special Objects' group in the **"Objects Window"**.

When a Log Window is inserted into a screen it will appear with its default size. After being inserted it can be resized as pleased by dragging the borders with the mouse.

The Log Window can display three different types of data: System Messages, Alarm Messages and Communication Driver Messages.



As many Log Window in as many screens can be inserted within the project according to your specific requirements.



The Event type associated to the Log Window can be dynamically changed during Runtime by using the object's appropriated Basic Script functions.

## Log Window Fields

---

The fields which must be presented in the Log Window can be selected by the programmer from those available by means of using the **"Field Choice"** window.

The fields or columns which can be displayed are described below:

### Event Text

The 'Event Text' column reports the description of the type of event which occurred. The event maybe a system message, a alarm text or a communication driver error, etc.

### Event Time

The 'Event Time' column reports the date and time of the data recording. The time is always in the operating system's local time and will be displayed with the "YYYY-MM-GG HH-MM-SS" formation.

### Event Id

The 'Event Id' Column reports the type of even which has been verified. For instance some event types are 'System', 'Driver', 'Alarm ON', 'Alarm OFF', etc. For further information please refer to the paragraph on "Log Window Fields".

### User

The 'User' column reports the name of the user active the moment the recording took place.

### Description

The "Description" column reports different information according to the type of event which occurred. For further information please refer to the paragraph on "Log Window Fields".

### Duration

The "Duration" column reports the event's duration. This field is only significant to some events. For further information please refer to the paragraph on "Log Window Fields".

### Event Num

The "Event Number" column reports the event's identification code number. For further information please refer to the paragraph on "Log Window Fields".

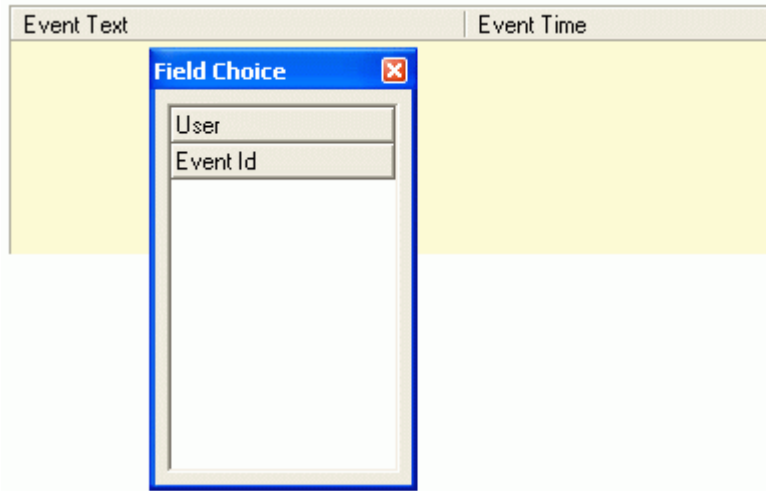
### Comment

The "Comment" column reports different information according to the type of event which occurred. For further information please refer to the paragraph on "Log Window Fields".

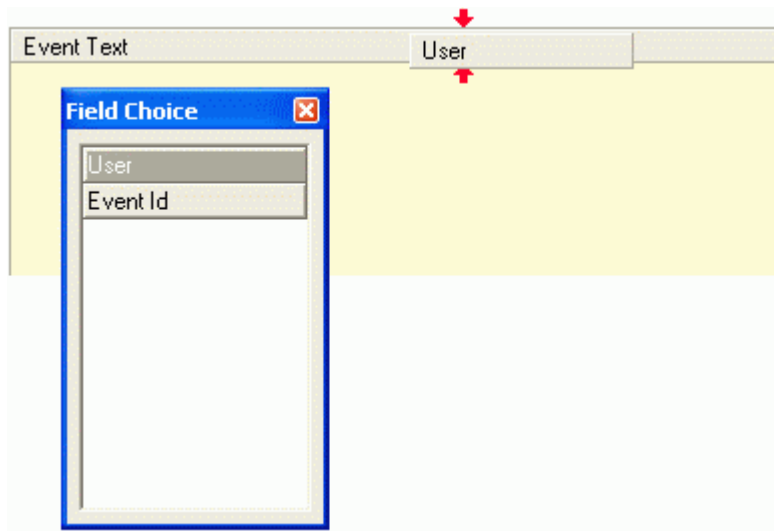


## Log Window Field choice

The Log Window object permits you to select the fields or columns to be displayed. In order to do this you need to open the **'Field Choice'** window by clicking on the **'Open'** button from the **'General'** group in the object's **'Properties Window'** or by double clicking and keeping the mouse button pressed on the object while pressing the **'SHIFT'** key at the same time.



The available fields which have not yet been inserted into the Log Window will be listed in the 'Field Choice' window. To move a field from the 'Field Choice' window to the Log Window just select the field with the mouse and drag it to the position desired in the Log Window. Note that the field can only be released at the beginning or end of an already positioned field, and that two red arrows should appear:



To move a field from the Log Window to the 'Field Choice' window just select the field with the mouse and drag it into the 'Field Choice' window.



When setting the window with the "Show Control Wnd" option during the Runtime you will be able to drag the columns out of the window where they will be deleted. You can, however, restore the initial configuration by using the "RecalcLayout" basic script method from the HisLogWndCmdTarget interface or when the page is reloaded.

## DataBase Fields

The DataBase of the Historical Log is composed of a certain number of fields, and not all of them are displayed in the Historical Log window. These fields can, in certain cases, be very useful for creating statistics or other things. For example, they can create reports by extracting data from the Database. There are three tables in the Historical Log file, SysMsgs (system messages), Alarms (alarm messages) and Drivers (communication Driver messages), but all these have the same fields, even though they may obtain different meanings from time to time, as explained below:

### TimeCol

This field reports the date and time of the recording in GMT (Greenwich Mean Time). The GMT is used to calculate the time in the rest of the world. The time zones are calculated by starting from Greenwich 0.00 hrs. Italy is "+1" hour ahead in respect to the GMT and "+2" hours at certain times of the year when the legal hour is put into force.

### LocalCol

This field reports the date and time of the recording taking the local time as reference.

### MSecCol

This field reports the milliseconds relating to the recording's time.

### UserCol

This field reports the name (Description or Electronic Signature) of the user logged on when the recording took place.

### EventCol

This field reports the event type which identifies the recording. The event types changed according to the table being consulted. The values may be:

Events	Event Num.	Table	Description
System	3006	SysMsgs	System Messages such as project startup, the log on/off of a user, etc.
Trace	3008	SysMsgs	This event type records a variable change whose " <b>Trace</b> " property has been enabled, in addition to the " <b>Add Msg to SysLog</b> " property
Driver	3005	Drivers	Messages relating to Communication Driver, such as communication activations, communication errors, etc.
ALARM ON	(*)	Alarms	Event recorded when the alarm is activated
ALARM ACK	(*)	Alarms	Event recorded when the alarm is acknowledged
ALARM OFF	(*)	Alarms	Event recorded when the alarm is deactivated
ALARM RESET	(*)	Alarms	Event recorded when the alarm is reset

(\*) alarm priority

### EvNumCol

This field reports an ID number code of the event type. The codes in question are reported in the table above. However, the ID number of the priority level set for the alarm will be recorded instead where Alarm tables are concerned.

**EvDescCol**

This field reports the description of the event. Where Alarm Tables are concerned, this field reports the text associated to the alarm threshold intervention.

In cases where the SysMsgs table has the Variable Trace function and the "Add Msg in SysLog" property enabled, it will show the name of the changed variable, its previous value, current value, changing value and the user who carried out the changes. An example of a message would be:

*The Variable 'VAR00001' (User '') has changed by ''. Previous value '7'. Actual Value '8'. Changing value '8'*

**DescCol**

In Alarm Tables, this field reports the name of the alarm threshold intervention and the variable associated to the alarm.

The operator's comment, entered with the appropriate window, will be saved in the SysMsgs table when the Variable Trace function and the "Add Msg in SysLog" property is enabled.

**CommCol**

In Alarm Tables, this field reports the alarm duration in text format.

The variable's description is saved in the SysMsgs table when the Variable Trace function and the "Add Msg in SysLog" property has been enabled.

**DurCol**

In the Alarm Tables, this field reports the alarm duration in numeric format (seconds).

**UnID**

This field shows the alarm's unique ID in the Alarms Table. This value is not shown in the "Historical Log Window", but is needed for managing the displaying of the "Alarms Historical" in the "Alarms Window" and in the "Historical Log Window".

**TraID**

This field shows the alarm's transaction ID in the Alarms table. This value is not displayed in the "Historical Log Window", but is needed for managing the displaying of the "Alarms Historical" in the "Alarms Window" and in the "Historical Log Window".

## Log Window Style Properties

---

The Style properties of the Log Window are used for setting the object's graphical properties.

To modify the Style properties, select the object with the mouse and use the Movicon **'Properties Window'**.

**Border**

The **'Border'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

**Clickable**

The **'Clickable'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

**Auto Layout**

The **'Auto Layout'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

**Show Control Wnd**

The **'Show Control Wnd'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

**Event Type**

The message type to be displayed in the Log Window is set in this option box. The options are:

- **System Messages:** only system messages will be displayed
- **Alarm Messages:** only alarm messages will be displayed
- **Communication Driver Messages:** only messages inherent to the Communication Drivers will be displayed
- **All:** All the three messages types, mentioned above, will be displayed



The event type to be displayed can be changed dynamically in the Log Window during Runtime by using the object's appropriated Basic Script functions.



**When the Historical Log is set for recording in IMDB, the "All" option in the "Historical Log Window" object will no longer be selectable. This is due to the fact that the IMDB (InMemoryDB) does not support the UNION clause (for the data extraction query from a number of tables).**

#### **Refresh Button**

The '**Refresh Button**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

#### **Filter Button**

The '**Filter Button**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

#### **Print Button**

The '**Print Button**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

#### **Report File**

The "**Report File**" property is described in the "Style Proprieties" paragraph on Drawings and Controls.

#### **Edit Report File**

The "**Edit Report File**" property is described in the "Style Proprieties" paragraph on Drawings and Controls.

#### **Button Size**

The '**Button Size**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

#### **Align Buttons**

The '**Align Buttons**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

#### **Refresh Button Text**

The '**Refresh Button Text**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

#### **Filter Button Text**

The '**Filter Button Text**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

#### **Print Button Text**

The '**Print Button Text**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

#### **Text Column Name**

The text which will appear as the name of the "**Event Text**" column is entered here. The default text will be used if this field is left empty.

#### **Time Column Name**

The text which will appear as the "**Event Time**" column's name is entered in this edit box. The default text will be used when this field is left empty.

#### **User Column Name**

The text which will appear as the "**User**" column's name is entered in this edit box. The default text will be used when this field is left empty.

**Event Column Name**

The text which is to appear as the name of the **"Event Id"** column is entered here. The default text will be used when this field is left empty.

**Description Column Name**

The text which is to appear as the name of the **"Description"** column is entered here. The default text will be used when this field is left empty.

**Duration Column Name**

The text which is to appear as the name of the **"Duration"** column is entered here. The default text will be used when this field is left empty.

**Event Num. Column Name**

The text which is to appear as the name of the **"Event Num"** column is entered here. The default text will be used when this field is left empty.

**Comment Column Name**

The text which is to appear as the name of the **"Comment"** column is entered here. The default text will be used when this field is left empty.

**Include milliseconds**

The **'Include milliseconds'** property is described in the paragraph on **"Style Properties common to Drawings and Controls"**.

## Log Window Execution Properties

---

The Execution properties of the Log Window are used for selecting from which project the data is to be taken.

To modify the Execution properties, select the object with the mouse and use the Movicon **'Properties Window'**.

**Project**

The name of the child project from where data is to be taken is entered in this edit box. If this field is left empty the current project will be used.



Only the name of an eventual child project of the current project is entered in this property.

**Max Rows**

The highest number of rows which can be displayed in the Log Window is entered in this edit box.

**Network Server**

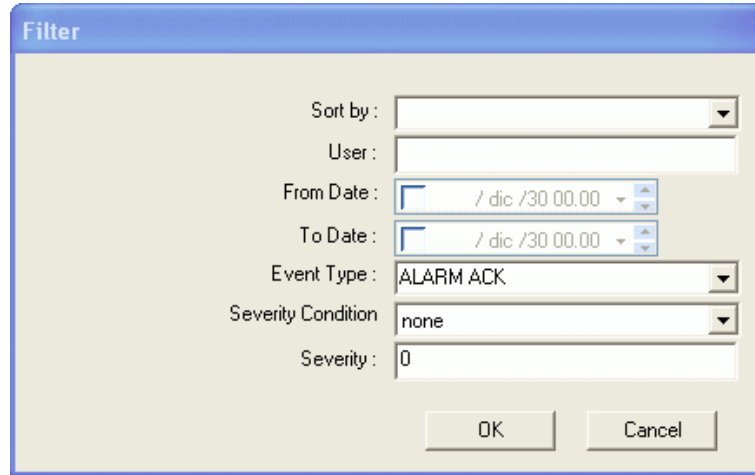
See paragraph "Drawings and Controls common Execution Properties".

**Network Backup Server**

See paragraph "Drawings and Controls common Execution Properties".

## Log Window Data Filter

Filters for displaying data in the Historical Log window can be executed during a project runtime mode. These filters can be done by following different data selecting criteria. When the **"Filter"** button, in the Historical Log Window, is pressed the following window will open for setting the filter:



When applying more than one filter at the same time you have to put "AND" between each one in the extraction query, then the conditions will be added up together.



The dialog window's character font and size can be customized by using the appropriate registry keys:

DBFilterFont  
DBFilterFontSize

Any modifications made to the font or its size will change the size of the dialog window.

### Sort by:

This selection is used for putting data into order according to the DataBase column chosen. The data order can be done by simply clicking on the column desired in the Historical Log Window, but only a few of the columns in the database are actually displayed. Therefore, if you want a 'sort by' based on the columns not displayed, you will have to use the filter window.

For further information on fields recorded in the DataBase please refer to the section on **"DataBase Fields"**.

### User:

This entry field is used for specifying the name of the user who wants to execute the filter. By entering the user's name, only data recorded during their log on will be extracted.



The text to be entered in this field is not the "Name" of the user, but their "Description or Electronic Signature".

### From Date:

The date with which the extractions are to start from is entered here.

### To Date:

The date with which the extractions are to finish is entered here.

### Event Type:

The selection is used for setting the event type the data must be extracted from. The items in the list may not have any meaning in certain cases. This depends on how the **"Event Type"** property from the Historical Log window has been set.

For further information on the event types available, please refer to the section on **"DataBase Fields"**.

**Severity Condition:**

This field is used for setting a compare condition for the severity of the alarms. This, for example, will ensure that only alarms with a specific severity are selected. This setting has meaning only when a severity value has been entered in the **"Severity"** field.

The possible values are:

- **"none"**: no compare condition will be applied
- **"="** alarms with the same severity as the one entered in the "Severity" field will be extracted.
- **">"** alarms with severities higher than the one entered in the "Severity" field will be extracted
- **"<"** alarms with severities lower than the one entered in the "Severity" field will be extracted

**Severity:**

This field is used for entering the severity value to used as a reference for extracting data. This setting only has meaning when a value different from **"none"** has been entered in the **"Severity Condition"** field.

# The Trends

*The Trends graphically show curve representations of variable behaviour and analysis of data filed by the recording engine.*

The Trends represent the most powerful tool for managing the displaying, analysis and logging of data contained on Movicon variables or on database files.

The Trends, in addition to their own intrinsic features and all the functional features of the Movicon drawings and vectorial symbols, are included with the Power Template© feature.

The Trend object is a tool which offers different ways of functioning, which not only display logged data independently but also permits the linking to database objects that have been inserted in the Data Logger. This allows logged data to be represented both on event or time.

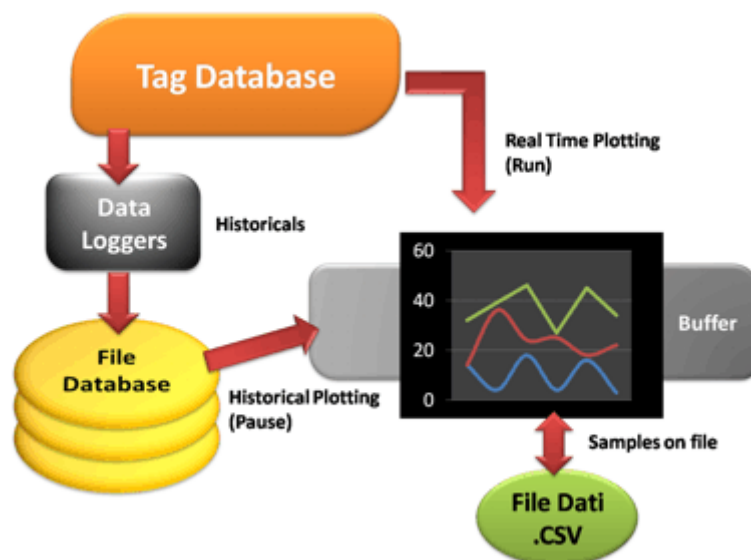
The Trend belongs to the Movicon "Trend - Charts - Data Analysis" category and therefore can be inserted in screens through the **"Objects Window"**.



The Trend is a powerful tool for displaying data recorded by Data Loggers and can at the same time record data in standard .CSV format.

The trend has been designed to guarantee maximum flexibility in managing graphical displays of recorded data. This means that this object can be used not only for representing data graphically but also as a recording function in its own right.

The Trend usually graphically represents data recorded by the Data Loggers, which carry out the task of the recording engines for the Trend. The Trend, however can itself record data of the variables associated in ".CSV" text format and retrieve data of values recorded with the data importer function. When the trend is kept inside an always active screen, due to the fact that the screen's "Not destroyable" option has been enabled, it will act as a constantly active recording engine.



A flowchart of how the Trends work.

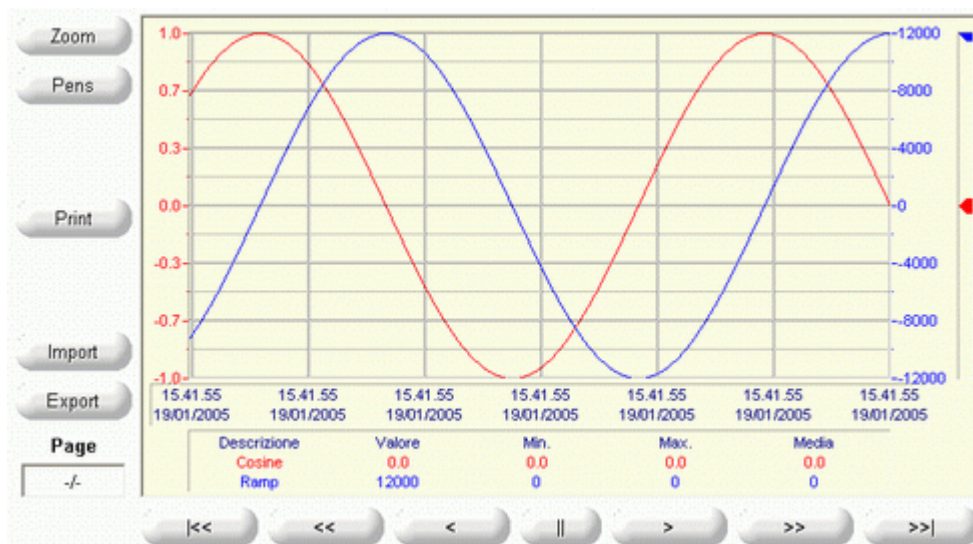
The Movicon Trends allow the behaviour of the plant's variables to be graphically displayed and recorded. The Trends are therefore the most efficient tool for filing, printing and analysing graphically the behaviour of variables which constitute the heart of the plant (usually analogic variables) such as temperatures, pressures, levels, scales, chemical measures or any other physical value can be converted into electronic signals.

Movicon also allows the managing of variable representations on Trend curve charts according to the different techniques which can be used.



## Time Based Trends

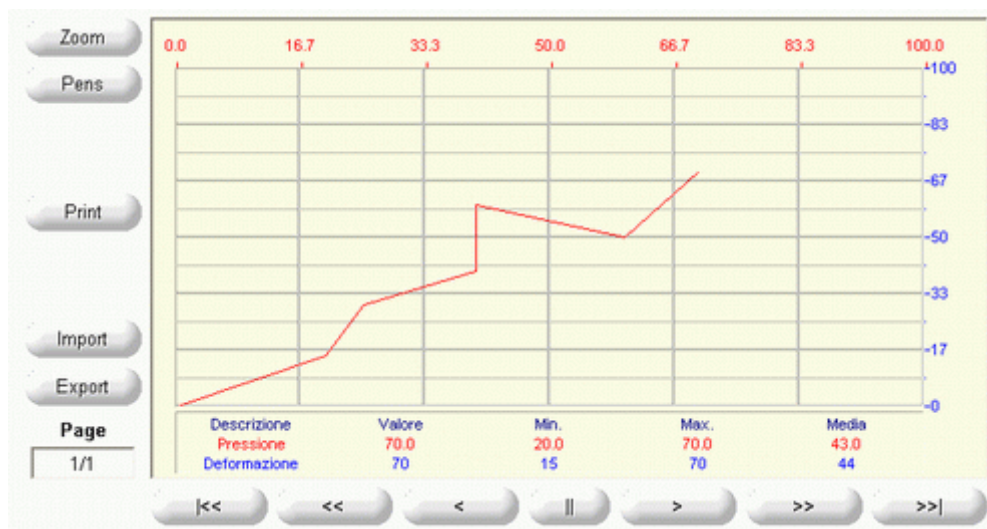
These types of trends are based on the Movicon vectorial drawings and allow a very advanced management of representing data and recording engines, as well as chart representations of database values recorded by Data Loggers.



This is an example of a template containing a symbol composed of a Trend object and other correlated button objects.

## X/Y Trends

The trend can also be used for displaying the behaviour of a curve shown on an X and Y chart, where the X axis is no longer associated to time but to the first pen on the list and the Y axis is associated to the second pen on the list. This is done in such a way that the coordinates of the X and Y points are determined by the associated variables, and a trace line is generated by the trend for each sampling which links the previous XY point to the current one. More XY curves can be displayed in one-only trend when other pen couples are inserted.



# Data Analysis

---

The Data Analysis is an object which derives from the Trend object, therefore it has many of the Trend's basic properties and also implements some new functionalities as well. The Data Analysis permits analysis to be executed exclusively on historical data, permitting you to have a powerful data analyzer graphics. The Data Analysis's new functionalities can be used differently compared to those used in normal Trends. Trends work in two ways, displaying data in runtime and displaying historicals (run-pause), whereas the Data Analysis permits you to use more advanced functions to carry out a deeper and more thorough analysis on historical data.

The Data Analysis object belongs to the Movicon "Trend-Charts-Data Analysis" category, from where it can be inserted on screen through the **"ToolBox Window"**.

## Data Analysis Functionality

---

The Data Analysis object is dedicated to analysing historical data consenting historical data to be viewed according to the different selectable date ranges, where the user can display data in different time scales. Data can be displayed in the following date ranges:

- minute range
- hour range
- day range
- month range
- year range

When the page is loaded the object will display the current date ranges: this means that if the hour has been set as the date range, the Data Analysis will load the data of the current hour when the page opens. For each prechosen date range you can scroll the database back and forth with the "<<", "<", ">" and ">>" buttons which change meaning (and title) according to the context chosen. For example, when choosing the month date range, the "<<" and ">>" buttons are used for scrolling the current month displayed back and forth, while the "<" and ">" buttons are used for scrolling individual days of that month back and forth.

The time range selection to be viewed is done by using the set of buttons at the bottom of the legend on the left (coloured green for default).

This object also consents you to execute comparisons of historical data being analysed graphically, by inserting a second curve for each pen using the buttons at the bottom right of the legend (coloured blue for default). The comparison curves will display the same time ranges (minute hour, etc.) but relate to the previous period selected using the appropriate comparison buttons. For instance, if you select the minute time range for the period to be viewed (using the green buttons), you will get the following results based on the comparison buttons selected (blue buttons):

**"(min)" Selection Button:** the minute time range is selected for viewing. By using the scroll buttons you can then select the minute you want to view. For instance, if "31-01-2008 10:15" is the data and time selected, the time range displayed will be "10:15:00 - 10:15:59".

**"None" Comparison Button:** no comparison will be activated.

**"(min.)" Comparison Button:** the minute prior to the one selected will be displayed on the comparison curve, therefore the "31-01-2008 10:14" minute.

**"(hour)" Comparison Button:** the same minute selected but relating to the previous hour will be displayed on the comparison curve, therefore the "31-01-2008 09:15" minute.

**"(day)" Comparison Button:** the same minute selected but relating to the previous day will be displayed on the comparison curve, therefore the "30-01-2008 10:15" minute.

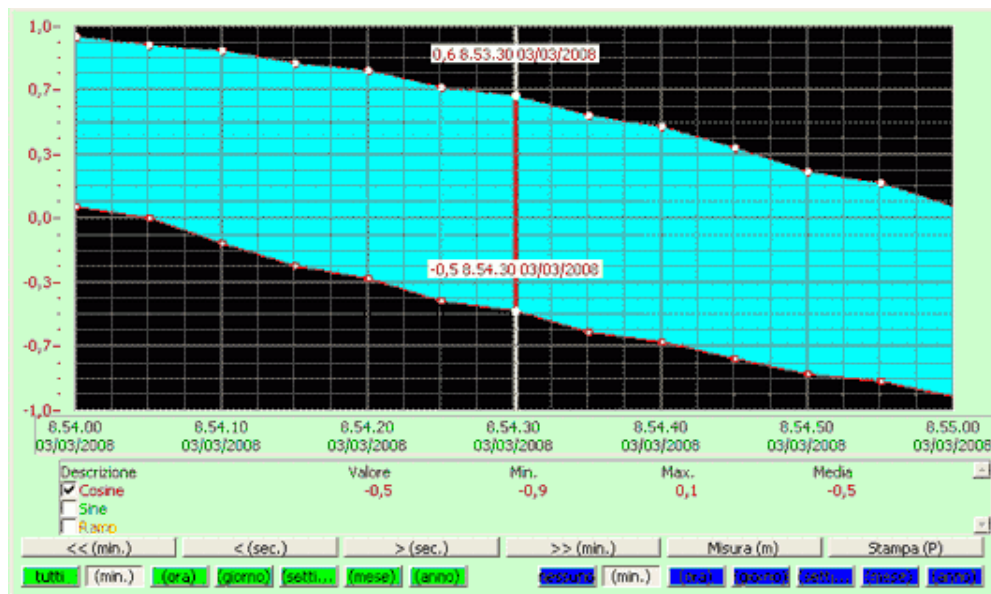
**"(Week)" Comparison Button:** the same minute selected but relating to the previous week will be displayed on the comparison curve, therefore the "24-01-2008 10:15" minute.

**"(month)" Comparison Button:** the same minute selected but relating to the previous month will be displayed on the comparison curve, therefore the "31-12-

2007 10:15" minute. When the month day selected is not in the previous month (ie. only a few months have a 31st day) the comparison curve will not display.  
**("year") Comparison Button:** the same minute selected but relating to the previous year will be displayed on the comparison curve, therefore the "31-01-2007 10:15" minute.

At this point, any differences between the two curves will be indicated by colouring in the area in question between them and by positioning the mouse cursor in this area, two labels will show reporting the two values and their date of recording. In this way you can confront one time range with the same previous time range or of an hour, day or week ago etc. To activate the compare function, as already mentioned, just click on one of the buttons from the "Compare Date Range Buttons" group.

The time range comparison selected must be consistent. If not data can be found to compare with this time range, the comparison curve will not display. For instance, when selecting the comparison curve for displaying data from the previous month that however has no data, the comparison curve will not display.



The above figure shows two curves, one displaying the selected minute range and the other comparing. The areas between the curves is coloured with a different colour and when positioning the cursor onto a sampling area, a label showing the corresponding values, recording date and time will show automatically for both curves. This type of comparison is simple and very immediate.

By using the "Measure" command you can get the X and Y difference of two points of one curve or of different curves (see the "Measure Button" property).

Furthermore, the Data Analysis consents to displaying only those values recorded by the Data Logger which have "good" qualities (this needs you to activate a specific option in the Data Logger column's "Add Quality" property).

### Custom Data Filter

During the runtime mode you can select a custom time range to view (different from the selectable time ranges using the appropriated buttons). This will allow you to display data from time ranges which suit your needs better such as 10 minutes, 1 day and 6 hours. To customize the time ranges simply click on the Data Analysis's "Time Area" to open the "Filter" window for setting the time ranges you wish to display. The items available are:

**From Date:** represents the data extraction start date and time

**To Date:** represents the data extraction end date and time

**From Date Compare:** represents the data extraction date and time for the comparison curve

**To Date Compare:** represents the data extraction end date and time for the comparison curve

There are no controls on selecting time ranges to be viewed in respect to those of the comparison curve. Therefore, make sure that you set the comparison curve time ranges to match those of the normal curve's time ranges so that the filter can work correctly. This can be done by verifying these conditions:

**To Date - From Date = To Date Compare - From Date Compare**

## Trend Functions

---

The Trend is a triple-functional tool:

1. Real-time variable curve viewer
2. Curve viewer of data recorded by Data Logger Data
3. Sampled output data recorder

The main job of the Trend is not just recording data files. This is usually done by the Data Logger (described in the appropriate section). The main job of the Trend is to represent data on a series of graphics by using the vast customizable function possibilities to make data clearer for the operator to understand.

The Trend can be used in recording data when it is desired to dispose data sampled by the object in output on file, typically readable by Ms. Excel or by any non database application. The ".CSV" format is used for data recorded in text format with separators which is interpreted by Excel and by many other applications.



The Trend, when enabled to record, files sampled data in standard ".CSV" format, but the recorded data is not automatically loaded into the Trend's buffer when opened. You can, however, consult this data by loading the saved file by using the appropriate Trend functions. The templates presented in the "Symbols Library" are already equipped with the "import" button in order for you to carry out this operation.

This type of recording is optional and must be enabled in the Trend object's configuration properties.

It is important to remember that when the Trend object is enabled to send sampled data in output on file, this will happen only when the object is active in memory, meaning when the screen containing it is displayed or has not been unloaded or destroyed with a change page.



Recording samples carried out by the Trend on ".CSV" file is executed by the object only when this is active in memory. To keep the Trend constantly active you need to make sure that the screen containing it does not get destroyed or unloaded from memory when using the change page function by checking the "Not Destroyable" option in the screen window's general properties.

The recording on file carried out by the Trend object is simply to be considered as an "output on file" of sampled values, totally asynchronous and separate in respect to any Trend link to the Data Logger. Any effective written data is performed through the settable cache memory so that the disk is accessed only when required according to the type of sampling carried out.

If a file recorded with Excel is opened, the values recorded on file will use the following the data sequence format:

	A	B	C	D	E
1	Time	Date	Cosine	Ramp	
2	15.05.52	27/01/2005	0.7	2400	
3	15.05.53	27/01/2005	-0.4	6400	
4	15.05.54	27/01/2005	-1	-9800	
5	15.05.55	27/01/2005	-0.2	-5800	
6	15.05.56	27/01/2005	0.8	-2000	
7	15.05.57	27/01/2005	0.8	2000	
8	15.05.58	27/01/2005	-0.2	5800	
9	15.05.59	27/01/2005	-1	9800	
10	15.06.00	27/01/2005	-0.4	-6400	
11	15.06.01	27/01/2005	0.7	-2400	
12	15.06.02	27/01/2005	0.9	1400	
13	15.06.03	27/01/2005	-0.1	5400	
14	15.06.04	27/01/2005	-1	9400	
15	15.06.05	27/01/2005	-0.5	-6800	
16	15.06.06	27/01/2005	0.6	-3000	
17	15.06.07	27/01/2005	1	1000	
18	15.06.08	27/01/2005	0	5000	
19	15.06.09	27/01/2005	-0.9	8800	
20	15.06.10	27/01/2005	-0.6	-7200	
21					

The date and time values are recorded according the operating system' settings in the International Settings item in the Control Panel.

The date separation character can be set in the Trend's configuration by selecting either Tab or Comma to suit the type of application being used.

## Trend Operability

The trend has two operating states, Start and Stop. These states are established in function with a variable which has been appropriately set for this purpose and assigned through the Trend's "Variables" properties.

When the Trend is set at Start, the data will be sampled and displayed as set in the execution properties assigned to the Trend. The buffer of data managed by the Trend can be sized as pleased and has a maximum capacity of 10,000 samplings.

When the Trend's status is switched to stop, the buffer's contents will be displayed. If the Trend is associated with a Data Logger, the buffer's contents may be a result of a data extraction from the database. Otherwise the buffer's contents will be the samplings carried out by the Trend in Start status.

Remember that also during the Stop status the Trend will continue to do samplings, buffer and record data.

The values in the buffer can be displayed by using the scroll commands which can be set in the Trend's properties and can be associated to Movicon variables. This will enable the cursor to point to data as specified in the operating configuration as desired and which will be described in further detail in the Trend properties.

## Linking Trends or Data Analysis to the Data Logger

The Trend and Data Analysis are pre-built to be linked to a recording engine of data managed in the project's Data Logger resource. The Data Analysis object is only capable of displaying data extracted from the Data Logger, whereas the Trend can manage two mode functions, Start and Stop. This makes it possible to manage Trend objects in 'Start' mode to view sampled data according to the modalities set, and in 'Stop' mode to view data extracted from the database, recorded by the Data Logger in function with the value extraction or order parameters associated to the query.

The trend, therefore can represent the extracted values, from the database file contents graphically, which is loaded when the Trend buffer switches to 'Stop' mode.

The Trend will be linked to the database by using the same link set for the Data Logger, and will execute the data order or filter by creating a RecordSet whose values will be automatically loaded in memory in the Trend's buffer when switched to Stop mode.



It is important to remember that unlike the Data Logger, the Default Query cannot extract values but can execute the SQL commands on them (Update or Insert). Extractions refer to the SQL "Select" commands managed from the "Default Filter" and "Default Order by".

## Displaying Trend data

---

Displaying data, represented in trend curves, within the Trend object can be greatly customized therefore made more powerful. The Trend sampler displays data in "Start" mode according to its sampling settings, which may also be different from those set for any associated Data Logger. In this case, in Stop mode, the Trend will display the values recorded according to the linked Data Logger's sampling settings. When the Data Logger records data and events or on variable value changes, the data will be same displayed by the Trend in "Stop" mode, but in this case the Trend will show a series of continuous dots or lines separated by Break Lines (if this property has been enabled) instead of a continuous curve line.

## Extended Functionality with Basic Script Interface

---

In addition to the usual animation functions Movicon allows a set of purposely preset commands to be associated to the Trend or Data Analysis object to be used in customizing functionalities in the object. These commands are available from the ActiveX Automation Members window in the 'Data Type' box under the VBA™ command set called **"TrendCmdTarget"**, which is accessed when editing the object's code through the "Script Explorer" window.



A large number of Templates of pre-configured multifunctional Trend objects can be found in the Movicon "Symbols Library".

Expert programmers should refer to the sections on the Basic Script Language for further information.

## Preconfigured Shortcut Keys

---

The Trend and Data Analysis objects are already preconfigured to manage some of the shortcut keys in automatic as described in the below table. When the Trend or Data Analysis has been selected with the mouse (clicking on object), thus receiving focus, pressing one of the shortcut keys will activate the relating command.

Keys	Command Description
0	Executes the "All" button function to display all available data (*).
1	Executes the "min." button function to display data for one single minute (*).
2	Executes the "hour" button function to display data for one single hour (*).
3	Executes the "day" command function to display data for one single day (*).
4	Executes the "week" button function to display data for one single week (*).
5	Executes the "month" button function to display data for one single month (*).
6	Executes the "year" button function to display data for one single year (*).

<b>CTRL+ALT+0</b>	Executes the "none" button function so that no data range is selected for the comparison curve (*).
<b>CTRL+ALT+1</b>	Executes the "min." button function for the comparison curve to select the comparison data for one single (*).
<b>CTRL+ALT+2</b>	Executes the "hour" button function for the comparison curve to select the comparison data for one single hour (*).
<b>CTRL+ALT+3</b>	Executes the "day" button function for the comparison curve to select comparison data for one single day (*).
<b>CTRL+ALT+4</b>	Executes the "week" button function for the comparison curve to select comparison data for one single week (*).
<b>CTRL+ALT+5</b>	Executes the "month" button function for the comparison curve to select data comparison data for one single month (*).
<b>CTRL+ALT+6</b>	Executes the "year" button function for the comparison curve to select comparison data for one single year (*).
<b>Freccia Sx/Dx o Su/Giù</b>	Move the cursor with in the graphic. Note: the point values nearest the cursor's position are shown in the legenda.
<b>PagUp</b>	Executes the "<<" button function. When the Trend object is in pause mode, this allows you to go to the next Trend page. As for the Data Analysis this depends on which time range view selected.
<b>PagDown</b>	Executes the ">>" button function. When the Trend object is in pause mode, this allows you to go to the next Trend page. As for the Data Analysis this depends on which time range view selected.
<b>S</b>	This is used for Run/Stop Trend types to switch over from one state to the other. Remember that the scroll and data analysis functions are enabled when in stop status (**).
<b>P</b>	This is used for printing out the page being displayed. A Windows interface opens for selecting the printer after which the print out operation can be confirmed
<b>M</b>	Executes the "Measure" button function. Enters and exits from the Measure mode (*).
<b>Z</b>	Executes the "Zoom" button function. Enters and exits from the Zoom mode.
<b>ESC</b>	This key has different functions according to the context in which it is used: <ul style="list-style-type: none"> <li>1. Exits from the Measure or Zoom mode when active</li> <li>2. Resets the Zoom applied to the graphic when already exited from Zoom mode.</li> <li>3. Cancels the data read operation and displays the graphic with data retrieved up to that moment.</li> </ul>
<b>F2</b>	Used for displaying the object in expanded mode or for returning it to its normal display size.

(\*) Functions available only for the Data Analysis object

(\*\*) Functions available only for the Trend object

## Mouse Executable Operations in RunTime

The Trend and Data Analysis provide commands in Runtime which can be activated with the Mouse to make data consulting operations much easier to carry out. These commands can also be managed through their respective "TrendCmdTarget" Basic script functions:

1. **Pen Configuration:** By double-clicking the mouse on a curve, legend pen or pen scale, the pen's configuration window will open (only if the pen has been set as editable in the design mode) in order for you to modify its name and variable properties. Modifications will be saved in the object's configuration file if predefined with one, otherwise they will be lost when the screen is closed.
2. **Curve Area:** By double-clicking the mouse on the area where the pen curves are traced, it will expand cutting off the parts showing the scales, legend and pens. The same command will restore the trend to its initial situation.
3. **Displaying Pens:** a single click on the check-box at the side of the legend pen's name will display or hide the curve of that pen. A CTRL+Click on this check-box will deselect all the pens presented and make only those clicked on visible.
4. **Scroll Trend:** keeping the right mouse key pressed down will able you to vertically pan the graphics (only in 'Stop' mode for Trends).
5. **Scroll individual Pens:** Pressing the "CTRL" key and the left mouse key will able you to pan each individual curve vertically. This is done by clicking on the curve of interest or clicking in the scale area of the pen desired.
6. **Selecting a Pen:** To select a pen to make the curve line thicker and highlight the pen in the legend, just "CTRL+Click" on the curve or click on the pen's name in the legend.
7. **Data filtering:** just clicking on the "Date-Time" area of a Data Analysis object will cause a "Filter" window to be loaded, allowing to set the time interval to show.

## Change Language for Pens

---

The Trend or Data Analysis Pens can also have change languages by inserting a string ID with the same name as the pen. The Trend or Data Analysis will display the text contained in the ID string instead of the pen's name.

## Interactive Legend Area

---

The Trend or Data Analysis object legend is interactive and is therefore possible to activate command by simply clicking or double clicking on the name of the pen. What happens when you click on pens:

The mouse **click** on the name of the pen (or on the pen's line) permits you to select the pen and get the following effects:

- Who whole pen line highlights within the legend
- The pen curve is highlighted in the chart. Corresponding curve is redrawn with a slightly thicker line in the chart compared to the others. Curves that are not "line" type are only displayed in the foreground and do not change thickness
- The scale associated to the pen is activated in cases where the pen is not visible. When the pen is no longer selected in the legend, the scale returns to its original state, therefore if originally invisible it will return to being invisible

**Double clicking** the mouse on the name of the pen (or on the pen's line) will open the pen's configuration window. This can also be done when double clicking on the pen's scale or curve. When a pen is set as "Not Visible" the pen's curve will no longer be displayed but the pen will remain on the legend's list in order to be made visible later on.

In addition to this you will find a checkbox on the left side of the pen's name which you can use to make the pen visible or invisible with a simple click, without having to revert to its configuration window every time needed. By keeping the "CTRL" pressed down while clicking on the pen's checkbox will de-select all the other pens except the one clicked on.



## Variable Quality Management

---

The Trend and Data Analysis objects obtain certain behaviours based on the qualities of the pens' variables associated to them. The function of these objects varies according to their type:

### Trend Object

When a variable has a quality different from "good" the pen and the name of the pen in the legend appear in grey (as disabled). In addition to this, when the Trend has not been associated with Enable Recording, the object will automatically be disabled when at least one of its variables has a quality different from "good". When disabled the Trend will no longer carry out any samplings, therefore its curves will no longer get updated. Any how, if the Data Logger is connected to the Trend, data will be loaded from the Data Logger when the Trend is stopped, including data recorded when the variable did not have a "good" quality.

### Data Analysis Object

This object gets data directly from the Data Logger connected. To manage data according to the variable's qualities you will need to enable the column in the DataLogger for recording the variable's qualities ("Add Quality" property). In this case the points that do not have 'good' qualities will not be joined in a line. When using a "rectangle" style for the pen, it will not be possible to check which are the points that do not have 'good' qualities graphically. On the other hand, when using an "area" or "line-area" style, the filling will be managed for all value, even those that do not have "good" qualities.

## Trend and Data Analysis Settings

---

The Trends and Data Analysis are objects belonging to the Movicon vectorial drawings and therefore have the same properties of vectorial drawings. In addition to the normal "Drawings and Controls Common Properties" forms, dealt with in the appropriate sections dedicated to them, The Trends and Data Analysis have a series of properties through which you can access specific settings of the Trend object. As the Data Analysis derives from the Trend object you will find that they have many of the same properties and some of their own which won't be available in the other.

## Trend and Data Analysis Style Properties

---

The Trend/Data Analysis Style properties are used to set the main style configurations of the window, whether graphical or functional.

To change the Style properties, select the object with the mouse and use the Movicon **"Properties Window"**.

### Border

The **"Border"** property is described in the paragraph headed **"Style properties common to Draws and Controls"**.

### Trend/Data Analysis Area Gap

The **"Gap"** property is used to customize, in pixels, the gap between the data and its border.

### Trend Type

This selection box allows you to set the trace scroll type:

- **Horizontal:** the values are scrolled horizontally in the Trend
- **Vertical:** the values are scrolled vertically in the Trend as a printout recording on paper
- **X/Y:** permits you to use the Trend as a XY chart. The X axis is no longer associated to time but to the first pen on the list. The Y axis is associated to the second pen on the list in such a way that the coordinates of the X and Y points are determined by the associated variables, and each sample creates a trace

which links the previous XY to the current one. Other XY curves can be displayed in the same Trend object by inserting other pairs of pens

#### **Trend/Data Analysis Title**

This check box permits you to display or hide the object's title which can be written in the objects "Title" property. The title is displayed on the Trend/Data Analysis top border.

#### **Show Break Lines Values Area**

This enabling box allows any zones of interrupted data recordings to be displayed. By doing this you will be able to see when and for how long the Data Logger did not record. When this property is disabled the data will be displayed without empty spaces even though in reality the curves show discontinuity.

These Break lines are never visible in the Trend in run because the Break Line Area's position and size can only be calculated with precision when the Trend loads the data from the Data Logger, therefore only when switching from run to stop mode, and not the other way round.



**The Break Lines are visible only when the Trend is in "Stop" mode.**

#### **Trend Mode**

This selection box is used for setting the Trends operating modality type:

- **Run-Pause:** this sets the Trend's operating modality to work on command or on event, in function with the state of the relating variable set in the "Variables" property group
- **Only Run:** this sets the Trend's operating modality to work in "Run" only, therefore the "Stop" (or Pause) analysis mode is not permitted. In this mode the Trend's values can represent the values only dynamically
- **Only Pause:** this sets the Trend's operating modality to work in "Pause" (or Stop), where it is not permitted to display data sampled in real-time but only representations of values loaded into the buffer by the query or the Basic Script commands

#### **Edit Pens**

When this command is activated a settings window of the **"Pen Properties"** assigned to the Trend/Data Analysis will open.

#### **Border**

This check box is used to enable the border relating to the Trend/Data Analysis's area to be displayed. This applies to the area displaying the curves only.

#### **Raised Border**

This check box is used to enable the border relating to the Trend/Data Analysis area, displaying the curves only, to be displayed with a raised 3D effect instead of sunken.

#### **Brush Style**

This property allows you to select the brush style to apply to the background of the Trend/Data Analysis's area displaying the curves only. The list shows various 'hatched' designs.

#### **Color**

This property allows you to select the background color to apply to the Trend/Data Analysis area displaying the curves only.

For further information on selecting colors please refer to the paragraph headed "Color Selections".

#### **Data Logger Link**

This selection box is used for specifying which Data Logger object the Trend/Data Analysis is to be linked to. This can be done by selecting one from the proposed list of Data Loggers inserted in the project.



After having selected the DataLogger from the list and confirmed this setting, Movicon will automatically add the pens to the Trend in function with the variables associated to the DataLogger.

**Refresh Pens from DataLogger Link**

This refreshes the trend's pen list by retrieving them from the linked DataLogger.

**Page Prev. Button**

This selection provides the command button for scrolling the previous Trend page. The command is only available when the Trend is in Pause mode.

This command changes its significance in the Data Analysis object according to the display interval type selected.

The same command can be executed with the "PagDown" key.

**Prev. Button**

This selection provides the command button for scrolling the previous Trend's recording.

This command changes its significance in the Data Analysis object according to the display interval type selected.

This command is only available when the Trend is in Pause mode.

**Pause-Run Button**

This selection provides the command button for switching over from Run status into Pause status and viceversa.

The same command can be executed with the "S" key.

**Next Button**

This selection provides the command button for scrolling the Trend's next recording.

This command changes its significance in the Data Analysis object according to the displayed range type selected.

This command is only available when the Trend is in Pause mode.

**Page Next Button**

This selection provides the command button for scrolling the Trend's next page. This command is only available when the Trend is in Pause mode.

This command changes its significance in the Data Analysis object according to the displayed range type selected.

The same command can be executed with the "PagUp" key.

**Expand Button**

This selection provides the command button for expanding the Trend/Data Analysis area to its maximum size. This command hides the scale, the pen area, the legend area, the buttons etc. and expands the display area of the curves to its maximum size. To return back to the previous mode use the "F2" key or double click the mouse on the Trend/Data Analysis area. The same command can be executed with the "F" key.

**Zoom Button**

This selection renders the Zoom command button available for zooming in and out of the Trend/Data Analysis area. You can use the "Z" key to do the same thing as well.

The Zoom command works in the same way:

Once this button is pressed, you can use the mouse to section to be enlarged. If at this point you wish to enlarge the curve even further, just select the part to be enlarged again.

To exist from Zoom mode in order to analyse data using the cursor, just press the Zoom button again, or press the "ESC" key. In this situation, where the zoom is no longer active, by pressing "ESC" key the curve will be restored to its original size. The following functions are active in Zoom mode:

- By keeping the Shift key pressed down, you can zoom the base times only once without modifying the pen scales.
- The zoom can be cancelled and the original graphic reloaded by tracing the zoom window backwards, from the bottom towards the top or right to left.

**Measure Button**

This selection makes the Measure Button available for carrying out measuring operations in the Data Analysis object. This command can also be carried out with the "M" key.

The measure mode consents to selecting a pair of points in the same curve, or different curves, and view the delta X delta and delta Y values on a ruler. The delta X value corresponds to the time ratio/difference between one point and the next, while the Y delta corresponds to value difference of the two points.

In order to select the two points after having activated the measure function, you must left mouse click on first point of the curve and by keeping the mouse key pressed down move

to the second point and then release. The two horizontal and vertical lines will remain visible showing the horizontal and vertical delta ratios between the two points.

#### **Print Button**

This selection provides the command button for executing a printout of the Trend/Data Analysis area. The same command can be executed with the "P" key.

#### **Show Date Range Buttons**

This selection renders the command buttons available for selecting the time ranges to be displayed in the Data Analysis object. These buttons are: all, (min.), (hour), (day), (week), (month), (year).

#### **Date Range Button Color**

This property consents you to select the color to be applied to the Data Analysis object's Date Range Buttons.

For further information on the color selections please refer to the "Color Selection" paragraph.

#### **Show Compare Date Range Buttons**

This selection renders the command buttons available for selecting the date ranges to be compared in the Data Analysis object. These buttons are: none, (min.), (hour), (day), (week), (month), (year).

#### **Compare Date Range Button Color**

This property consents you to select the color to be applied to the Compare Data Range buttons in the Data Analysis object.

For further information on color selections please refer to the "Color Selection" paragraph.

#### **Date Range**

This property consents to selecting a data range to be displayed for default when loading the Data Analysis object. The possible selections are: all, (min.), (hour), (day), (week), (month), (year).

#### **Button Size**

The '**Button Size**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

#### **Align Buttons**

The '**Align Buttons**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

#### **Page Prev. Button Text**

The text which is to appear on the '**Page Prev. Button Text**' is entered in this edit box. The default text will be used if left blank.

#### **Prev. Button Text**

The text which is to appear on the '**Prev. Button Text**' is entered in this edit box. The default text will be used if left blank.

#### **Pause-Run Button Text**

The text which is to appear on the '**Pause-Run Button Text**' is entered in this edit box. The default text will be used if left blank.

#### **Next Button Text**

The text which is to appear on the '**Next Button Text**' is entered in this edit box. The default text will be used if left blank.

#### **Page Next Button Text**

The text which is to appear on the '**Page Next Button Text**' is entered in this edit box. The default text will be used if left blank.

#### **Expand Button Text**

The text which is to appear on the '**Expand Button Text**' is entered in this edit box. The default text will be used if left blank.

#### **Zoom Button Text**

The text to appear on the "**Zoom Button**" is entered in this edit box. The default text is used when this field is left blank.

### Measure Button Text

The text to appear on the "**Measure Button**" is entered in this edit box. The default text is used when this field is left blank.

### Print Button Text

The text which is to appear on the '**Print Button Text**' is entered in this edit box. The default text will be used if left blank.

### Default Filter

This edit box permits you to enter a text string containing the filter function in SQL standard language on data to be selected from the Database. The filter functions automatically shows the WHERE suffix in a drop-down window, allowing you to simply indicate the filter parameters.

### Default Sort

This edit box allows you to enter a text string containing the sort function in SQL standard language in data to be selected from the Database. The sort functions automatically drops down a window with the ORDER BY suffix in it, allowing you to simply indicate the data sort parameters.

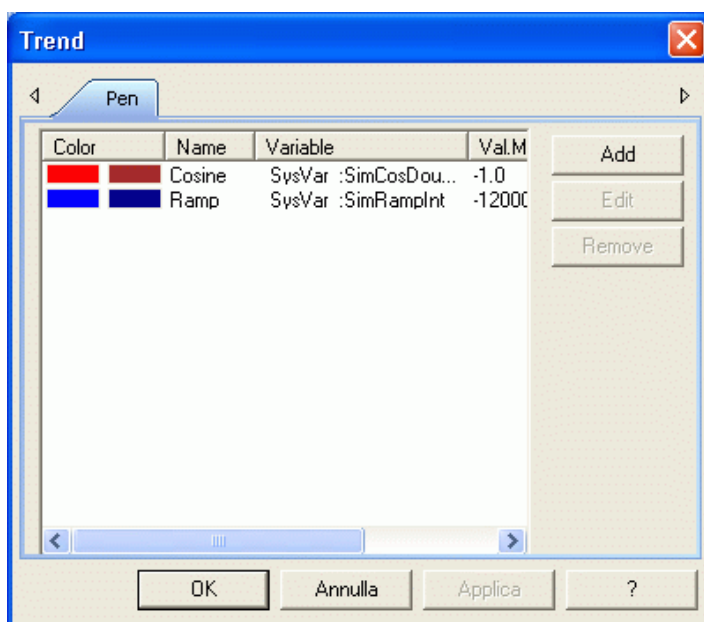
### Default Query

This edit box allows you to enter a text string containing the query in **SQL** standard language in data to be selected from the Database.

## Trend and Data Analysis Pen Properties

The Pen properties are used for setting associations between the curves displayed in the Trend/Data Analysis and the Movicon variables, being the same variables of the associated Data Logger.

Each pen accesses to a following window for setting the complete chart style desired.



This window shows the pen table inserted in the objects, indicating the colors, names and database variables associated.

To insert new pens you need to press the **Add** button on the right hand border, through which the setting window, as shown below, is activated.

To change the pen settings previously inserted you need to activate the **Edit** button after having selected the pen desired.

To delete a pen from the object, you have to select it first, then press the **Remove** button.

## Pen Configurations

---

The **Edit Pen Properties** window is accessed when the **Add** or **Edit** commands are used. This window can also be accessed in RunTime by double-clicking on the pen line you wish to change. However, the pen's name of the variable associated to it cannot be changed when accessed in this way. Furthermore, if a configuration file has not been associated to the object, any changes made will only be kept until the Trend remains active and will be lost when the screen is closed.

The Edit Pen Properties are subdivided in groups for graphic reasons only.

### General

#### Name

The name you wish to assign to the pen is entered in this box. The name can also be different from the variable's and will be represented in the Trend/Data Analysis window's legend and recorded on file as the name of the assigned values.



The text string which can be associated to the pen supports all the special and space characters, leaving it to the programmer's discretion to use the separation characters (tabulations or commas).



The Trend/Data Analysis's pens can also have the change language function by inserting a string ID with the same name as the pen's.

#### Variable

The name of the variable to be associated to the Trend/Data Analysis's pen is entered in this edit box (or selected with the "..." browse button on the right). The selected variables will be sampled by the Trend according to the modalities set.

New variables can be entered into the Real-Time DB if needed by double-clicking on the option box.

#### Data Logger Col.

The pen's name can be different to the name of the Data Logger's column. The "Data Logger Col." property allows you to set the reference column for the pen and therefore assign any name as "Pen Name". When no name is set, the column's name will be used instead.

#### Visible (Pen)

This selection box permits the trace visibility to be set in the Trend/Data Analysis window, allowing data to be sampled without displaying the values in the Trend. The Pen name will however remain visible in the Legend even when the curve is not displayed in the Trend/Data Analysis.

#### Visible in Legend

This option allows the pen to be displayed or not to be displayed in the Trend/Data Analysis legend.

#### Editable

This property allows you to make the pen editable during Runtime mode. A pen can be edited during Runtime by double clicking on the curve or the pen's name in the Legend or on the pen's scale.

### Scale



Pen scale major tick values can be customized using the string table. By inserting a string identifier in the string table with this format: "<PenName><Tick Number>" (where the tick number starts from '1'), the value represented in a pen scale tick can be replaced by the text desired.

**Auto**

When this property is enabled the pen's scale will automatically adapt according to the value obtained by the pen so that the highest value obtained by the pen is displayed at the top of the scale.

The automatic scale is managed in this way to guarantee that the first and last ticks are always set with the integer values without decimals.

**Min. Value**

The lowest value which the variable in the Trend/Data Analysis may obtain is set in this edit box and which will be returned on the variable's scale situated on the Trend/Data Analysis's border at the side.

**When the variable linked to the Trend/Data analysis has its "Enable Scaling" property enabled, the minimum value used will be the one set in the variable's "Scale Min." property.**

**Max.Value**

The highest value which the variable in the Trend/Data Analysis may obtain is set in this edit box will be returned on the variable's scale situated on the Trend/Data Analysis's border at the side.

**When the variable linked to the Trend/Data Analysis has its "Enable Scaling" property enabled the minimum value used will be the one set in the variable's "Scale Max." property.**

**Format**

By using this option box you can set how the numeric format of the variable associated to the Trend is to be displayed.

The types of format available are represented by syntax:

**x:** where the number of x identifies the figure to be displayed

**x.x:** where the number of x after the decimal point indicated the decimal figure to be displayed

Formats with decimal figures, "**x.x**" have meaning only when a variable is inserted in 'floating comma' format. When the number is integer the decimal figure will always remain at zero.



In case where this property has not been set (left blank) the Trend/Data Analysis will inherit any format defined in the associated variable's "Default Format" property.

**Visible (Scale)**

When checking this box the scale of values will be enabled on the border at the side of the Trend/Data Analysis window. If left unchecked, the scale will not be displayed for pen being configured.

**Right-Bottom**

When this property is enabled the pen's scale will be displayed at the bottom or on the right of the Trend/Data Analysis area.

**Log. scale**

When this property is enabled, the scale displayed for the pen will assume the logarithmic values instead of the linear values for default.

**Eng. Unit**

This edit box is used for inserting a text for identifying the Trend/Data Analysis's variable's engineering units to be represented. In instances where a text is not entered, the one in the variable will be used instead. The engineering unit will be displayed in the legend after the pen's description.

**Style****Plot Type**

The pen configuration provides the possibility to select the type of trace to be assigned to the variable, by choosing one of the four options offered by the system on the list available.

The curve graphic possibilities are:

- **Line:** the trace is represented by one simple line
- **Rectangle:** the trace is represented by a series of vertical bars (bargraph)
- **Area:** in this case the area subtended by the chart's curve is floodfilled by a color
- **Line-Area:** in this case the area subtended by the chart's curve is filled with a series of vertical bars

#### **Pen Color**

By using the "Color" selection button you can set the color for the pen, its trace in the Trend/Data Analysis and any correlated data.

For further information on selecting colors please refer to the paragraph on "Color Selection".

#### **Pen Size**

This edit box permits you to size the line (in pixels) associated to the variable.

#### **Pen Style**

This is used to set the trace's graphical property. By means of using the list you can select the type of line to be displayed which may be a continuous solid line, dashes or other.

#### **Brush Style**

This is used for setting the graphic style for the area subtended by the trace in the Trend/Data Analysis, when a non standard line type has been selected.

#### **Brush Color**

This is used for setting the color for the Trend/Data Analysis trace's floodfill, when a non standard line type has been selected.

For further information on selecting colors please refer to the paragraph on "Color Selection".

#### **Points**

This setting highlights the points along the Trend/Data Analysis curve.

#### **Labels**

This setting permits the pen's values to be displayed in correspondence to each point in the graphic. Labels are only controlled in the Trend object and which are displayed near each point only if these points are not too near to each other.

#### **Saved Values**

This setting allows the curve to be frozen upon enabling this property. The frozen curve is then displayed in the Trend with a dashed line while the actual curve will continue running in the Trend. Therefore, two curves will be obtained for the same pen, one in realtime and the other one saved the movement this property was enabled. This function is used for comparing curve values in realtime with those from a moment ago. When this property is disabled again, the frozen curve will be cancelled and on the one in realtime will remain in the Trend.

Enabling/disabling this property only has significance in runtime mode only.

### **Statistic**

#### **Minimum Pen / Maximum Pen / Average Pen**

When these properties are enabled, an additional trace will be displayed in the Trend/Data Analysis window, set with a standard horizontal line representing the Minimum value, the Maximum value and the Average value assumed by the variables associated to the pen, among those sampled and filed in the Trend's buffer.

When enabled, these traces can be set with a color, size and style as described below.

#### **Minimum / Maximum / Average Pen Color**

The colors of the traces representing the Minimum, Maximum and Average values assumed by the variable associated to the pen are set by using the 'Color' selection button.

For further information on selecting colors please refer to the paragraph on "Color Selection".



**Minimum / Maximum / Average Pen Size**

This edit box is used for setting the size of the trace representing the Minimum, Maximum and Average values assumed by the variable associated to the pen.

**Minimum / Maximum / Average Pen Style**

This is used to set the graphical property of the traces representing the Minimum, Maximum and Average values assumed by the variable associated to the pen. By means of using the list you can select the type of line to be displayed which may be a solid line, dashes or other.

## Trends Pen Area Properties

---

The Pen Area Properties of the Trend are used for setting up the main configurations of the area in which the pens, indicating the real-time values of the variables associated to the Trend, are displayed.

To change the Pen Area properties, select the object with the mouse and use the Movicon **"Properties Window"**.

**Visible**

The **"Visible"** property allows the Trend's "Pen Area" to be displayed or hidden.

**Border**

The **"Border"** property allows the border relating to the Trend's "Pen Area" to be displayed. This applies to the display area of the pens only.

**Raised Border**

The **"Raised Border"** property allows the border relating to Trend's "Pen Area" to be displayed with a raised 3D effect instead of sunken. This applies to the display area of the pens only.

**Size**

This property is used for setting the size of the Trend's "Pen Area" in pixels. This applies to the display area of the pens only.

**Brush Style**

This property allows you to select the type of brush stroke (hatched) from the list of the various styles available, to be applied to the Trend's "Pen Area", being the display area of the pens only.

**Color**

This property allows you to select the background color to be applied to the Trend's "Pen Area", being the display area of the pens only.

For further information on the color selection please refer to the paragraph on "Color Selection".

## Trend and Data Analysis Legend Area Properties

---

The Trend/Data Analysis Legend Area properties determine the main configurations of the area in which the legend of the Trend/Data Analysis's pens are displayed.

To change the Legend Area properties, select the object with the mouse and use the Movicon **"Properties Window"**.

**Visible**

The **"Visible"** property consents the Trend/Data Analysis's "Legend Area" to be hidden or displayed.

**Max. Nr. Visible Pens**

This property allows a limit to be set on the number of pens displayed in the legend so that it will not get oversized. Small Scroll buttons will appear to the right of the legend to scroll the legend's pen list. This property has been preset with the value '4'.



The Trend objects will displayed with these characteristics when opening projects created with the previous Movicon version.

The "MaxLegendVisiblePen" property from "TrendCmdTarget" basic interface allows this parameter to be read/changed in runtime. In this case, when modified a "Refresh" will need to be carried out to update the Trend/Data Analysis graphically with the new settings.

**Border**

The "**Border**" property consents the border relating to the Trend/Data Analysis's "Legend Area" to be displayed. This applies to the display area of the legend only.

**Raised Border**

The "Raised Border" property allows the border relating to the Trend/Data Analysis's "Legend Area" to be displayed with a risen border instead of a flat one. This applies to the display area of the legend only.

**Brush Style**

This property allows you to select the type of brush stroke (hatched) to be applied to the Trend/Data Analysis's "Legend Area". This applies to the display area of the legend only.

**Color**

This property allows you to select the background color to be applied to the Trend/Data Analysis's "Legend Area". This applies to the display area of the legend only.

For further information on the color selection please refer to the paragraph on "Color Selection".

**Font**

A window will activate by using the "..." button on the right hand of the box where you can select the font to be associated to the texts representing the Trend/Data Analysis's legend. The selection is done by using the Windows standard modalities.

## Trend and Data Analysis Time Area Properties

---

The Trend/Data Analysis's Time Area properties determine the main configurations of the area in which the data and time, corresponding to the major divisions of X axis, will be displayed.

To change the Time Area properties, select the object with the mouse and use the Movicon "**Properties Window**".

**Visible**

The "**Visible**" property allows the Trend's "Time Area" to be displayed or hidden.

**Automatic Scale**

When enabled this option consents the date range scale axis to adapt to the data loaded in the Data Analysis object accordingly. When enabling this option, the buttons for scrolling data ("<", "<<", ">" and ">>") and inserting comparison curves become no longer usable.

**Border**

The "**Border**" property allows the border relating to the Trend/Data Analysis's "Time Area" to be displayed. This applies to the display area of the date and time only.

**Raised Border**

The "**Raised Border**" property allows the border relating to Trend/Data Analysis's "Time Area" to be displayed with a raised effect instead of flat one. This applies to the display area of the date and time only.

### Brush Style

This property allows you to select the type of brush stroke (hatched) from the list of the various styles available, to be applied to the Trend/Data Analysis's "Time Area", being the display area of the data and time only.

### Color

This property allows you to select the background color to be applied to the Trend/Data Analysis's "Time Area", being the display area of the time and date only. For further information on the color selection please refer to the paragraph on "Color Selection".

### Text Color

A window will activate by using the "..." button on the right hand of the box where you can select the color to be associated to the texts representing the Trend/Data Analysis's time area.

For further information on the color selection please refer to the paragraph on "Color Selection".

### Font

A window will activate by using the "..." button on the right hand of the box where you can select the font to be associated to the texts representing the Trend/Data Analysis's time area. The selection is done by using the Windows standard modalities.

### Font Orientation

This property tilts the display of the strings which represents the recording's date and time. The maximum tilt is 45 degrees. This allows the number of vertical label to be increased without superimposing any texts.

### Show Ms

This property permits the values (in milliseconds), in recording time of the Trend/Data Analysis's "Time Area", to be displayed or hidden.

### Show Date

This property permits the recording date, displayed in the Trend/Data Analysis's "Time Area", to be displayed or hidden.

### Time Format

This property consents to formatting dates and times with which the data is to be displayed in the Trend/Data Analysis time area. The "**Show Ms**" property will remain unchanged and continue to add milliseconds. The "**Show Date**" property however, does not have any effect in cases where this custom format is used.



**When the date and time format has been customized, it will be represented on one line only.**

All the format codes that can be used in the property are:

Code	Significance
%a	Name of week day abbreviated
%A	Name of week day in full
%b	Name of month abbreviated
%B	Name of month in full
%c	Local date and time
%d	Day of the month in numeric form (01-31)
%H	Time expressed in the 24 hour format 24-ore (00-23)
%I	Time expressed in 12 hour format (01-12)
%j	Day of the year in numeric form (001-366)

%m	Month in numeric form (01-12)
%M	Minutes in numeric form (00-59)
%p	A.M./P.M. Indicator for 12 hour displays
%S	Seconds in numeric form (00-59)
%U	Week of the year in numeric form, with Sunday being the first day of the week (00-53)
%w	Day of the week in numeric form (0-6; Sunday = 0)
%W	Week of the year in numeric form, with Monday being the first day of the week (00-53)
%x	Local Date
%X	Local Time
%y	Year without century in numeric form (00-99)
%Y	Year with century in numeric form
%z,%Z	Time zone abbreviated or in full
%%	Percent sign

## Trend and Data Analysis Grid Properties

The Trend/Data Analysis Grid properties are used to configure the identical styles of the Trend/Data Analysis's X and Y axis.

To change the Grid properties, select the object with the mouse and use the Movicon **"Properties Window"**.



The grid is subdivided into two subdivision types, Major and Minor, meaning the thickness of the lines subdividing the grid fields.

The grid is represented with double colour semi-transparency. This allows you to quickly single out areas between two major ticks. To deactivate the double colour transparency effect, set the option for displaying the grid in normal mode ("Normal X/Y Grid" options).



**The transparency effect is not managed by runtime for Windows CE.**

### XY Grid Visible

These option boxes permit the grid's major lines to be displayed or hidden for the X axis (horizontal axis) or for the Y axis (vertical axis).

### X/Y Minor Grid Visible

These option boxes permit the Grid's Minor lines to be displayed or hidden for the X axis (horizontal axis) or for the Y axis (vertical axis). Only has effect with the grid's Major lines enabled.

### Nr. Major X/Y Ticks

This edit box is used for customizing the number of Major lines, by entering a value from 1 to 100, to be displayed for the grid in the Trend/Data Analysis window whether for the Grid X (horizontal axis) or the Y Grid card (vertical axis).

### Nr. Minor X/Y Ticks

This edit box is used for customizing the number of Minor lines, by entering a value from 1 to 100, to be displayed for the grid in the Trend/Data Analysis window whether for the Grid X (horizontal axis) or the Y Grid card (vertical axis).

### **Logarithmic X/Y Grid**

When enabling this check box, the X or Y grid will be displayed according to the logarithmic functions, instead of in linear mode as for default.

### **Normal X/Y Grid**

When enabling this check box, the lines in the X or Y grid will be displayed with a straight line instead of a dashed line as for default.

### **X/Y Grid Color**

By using the standard color selection, you can assign the colors desired to the grid's lines whether being for the X Grid card or (horizontal axis) or for the Y Grid card (vertical axis). For further information about color selections please refer to the paragraph on "Color Selection".

### **Grid Overlap**

When enabling this check box the Trend's grid will overlap to the curves.

### **Grid Shifting**

When this check box is enabled the grid will move in the Trend area together with the variable curve lines during runtime. When this box is left unchecked only the curves will move while the grid remains still.

The grid's shift is not active with XY charts.

## **Trend and Data Analysis Execution Properties**

---

The Execution properties are used for setting Trend/Data Analysis's data samplings. To change the Execution properties, select the object with the mouse and use the Movicon "Properties Window".

### **# Samples**

This edit box allows you to set the highest number of samples to be managed by the Trend's sampler and therefore establishing the buffer's memory capacity of the object. When this value is reached while sampling, the oldest data exists from the buffer to make place for new data (FIFO).



10,000 is the highest number of samples that can be managed when the Trend is configured as "Horizontal" or "Vertical". However, when the Trend has been setup as "X/Y", the highest sampling number is limited to 1000 values to keep the user interface light otherwise it will become very slow to use if exceeded.



This value is also used for the memory capacity for loading values of any query executed when the Trend is put into Pause (stop) mode. Extracted values exceeding this limit will not be managed.

For the Data Analysis object, this value defines the maximum limit for the number of points which can be shown. To get the real number of points, the property value should be multiplied by 10. Being the property value set to 3600 means that the Data Analysis object can extract  $3600 \times 10 = 36000$  samplings. This limit is to avoid memory waste and should be set to a value suitable for the computer memory performances. Setting a too high value can cause an "Out of memory" error while getting data from the database.

The Web Client uses the value set in "Max Rows" field multiplied by 10 (default  $1000 \times 10 = 10000$ ). This allows to have different values for a server with respect to a Web Client or a Network Client.

### **Record Every**

This property is used for setting the sampling time desired for the Trend. The values can be set in Hour, Minute, and Second time units.

The resulting value will be the one used for sampling data which is buffered in memory and displayed according to the modalities set.

### **Record Every (ms)**

This property is used for setting the sampling time in milliseconds.



As you can see, Movicon also allows sampling times to be set in milliseconds, which will require a more demanding CPU performance of the PC which is the programmer's responsibility to evaluate.

#### **# Update Every**

This edit box is used for setting the Trend object's video refresh time of data sampled by the Trend. The number set corresponds to the number of samples executed before the video refresh of sample values is carried out.

The difference between the values being acquired and the values being displayed may be useful to optimize the resource's task without effecting high precision sampling performances.

#### **View Timeframe**

This edit box is used for setting the timeframes to be viewed in the Trend area, whether in Start or Stop mode.

The timeframe set will determine the number of samples, indicated in the "# Visible Samples" box, to be viewed.

There is no limit on the amount of time that can be set and that the Trend can display on any one page. Therefore any time can be set in "dd:hh:mm:ss" which will be used by the Trend in runtime to view each of its pages. Any recording "holes" on the data logger will be represented with grey zones (without data), and not with lines. In this way a Trend page will always display a data interval equal to the time set in the "View Timeframe" property.

The Trend used its specific algorithm to display a lot of data in one page only where curves will be represented without altering their appearance with approximations.

#### **# Visible Samples**

This read only property shows the number of samplings to be displayed in the Trend area.

This value depends on the Trend's window size in pixels and the "View Timeframe" settings.

#### **Record On File**

The Trend's data recording function is activated when this selection is enabled. In this case, the object will record the data it has sampled on file in ".CSV" format according to the sampling modalities set.



The recording of samples executed by the Trend on ".CSV" files is executed by the object only when this is active in memory. In order to keep the Trend object always active, you have to make sure the screen, it is contained in, does not get destroyed and unloaded from memory with a change page by checking the "Not Destroyable" option in the screen's general properties.

If this property is enabled when a Trend is opened, the data is read from a linked text file each time the Trend is paused.

#### **File Name Suffix**

The name you wish to use for recording the output of data sampled by the Trend object must be declared here. The data will be recorded on file in standard .CSV format only if the "Record On File" has been enabled as described above. The date and time of the recording will be added to the suffix if the 'New File' property is enabled.

#### **# events cache**

This edit box is used to configure the number of samples to be buffered in memory before being unloaded on file. This value can be set in function with the sampling time, and is needed to avoid over-accessing the RAM disk with high-frequency samplings.

When the number of samplings indicated in the box has been reached, the system will save the recordings from the disk memory cache.

#### **Max. File Length**

This property is used to set the file's maximum length in KBytes. When this size is reached the data will be recycled or a new file will be created according to the settings carried out.

#### **Max. # Files**

This property is used for setting the number of files to be created when the "New File" property is enabled. When this number has been reached the data will be recycled starting with the oldest file.

### **New File**

This selection determines the behaviour of output recordings on file of sampled data at the recording's startup. Recording starts when the trend object (or display screen or project startup) is reloaded in memory. At the start of the project process with this box enabled, a new file will be created and the previous data will be cancelled by being rewritten on the old file. When this function is disabled, the data will be recorded in line with the old data on the same file. When a number higher than one has been set in the "Max. # Files" property, a new file will be created at the start of the object's process.

### **Network Server**

This property allows you to insert the name of the Movicon network server. In this way the Trend/Data Analysis can ask the Movicon Server for any historical data it has to retrieve and not by using the local project Data Logger's DBMS connection. Historical data from a remote Data Logger will then be displayed on the PC in Trend/Data Analysis form. Seeing that the Trend/Data Analysis is able to retrieve historical data from a remote Data Logger, the following properties will need to be set in the Trend object:

- Execution -> Network Server
- Execution -> Max Rows
- Style -> Data Logger Link

The data logger connected to the Trend/Data Analysis must also be configured in the local project. However, its "Enable" property can be disabled to prevent any recordings taking place in local database.

### **Network Backup Server**

See paragraph "Drawings and Controls common Execution Properties".

### **Max Rows**

This property is used for setting the number of records to be loaded from the remote Data Logger when the "Network Server" property is active.

### **Commands on Click**

See paragraph "Execution Properties" in Drawings and Controls.

### **Ext. File Settings**

Name of "Configuration Files" in which the Trend/Data Analysis's configuration settings are saved or loaded during RunTime.

The configuration file, set in this property, is automatically loaded when the Trend/Data Analysis is displayed, and automatically saved when any modifications are done to its pen's properties through the appropriate window.

See paragraph "Execution Properties" for Drawings and Controls.

### **Generate Unique File Name**

See paragraph "Execution Properties" for Drawings and Controls.

### **Load**

By using this command the configuration file settings selected in the "Configuration File" property will be loaded and applied to the Trend/Data Analysis object.

See paragraph "Execution Properties" for Drawings and Controls.

### **Save**

By using this command all the Trend/Data Analysis object's settings in the configuration file selected in the "Configuration File" property will be saved.

See paragraph "Execution Properties" for Drawings and Controls.

## **Trend Variables Properties**

---

The Variables property form is used for configuring the variables for the Trend's operation commands.

Be reminded that the Trend's buffer contents, when switched to Pause (Stop), are determined by the result of a query in the Data Logger (if present), or from data sampled by the Trend's engine.

To change the Variable properties, select the object with the mouse and use the Movicon **"Properties Window"**.

**Enable Recording**

The name of the variable to be associated for enabling and disabling the sampling of data associated to the Trend pens, is entered in this edit box (variable name can also be selected with the "..." browse button on the right).

Enabling the sampling (and recording) of values is independent of the Trend's operating status.

The "0" value of the variable associated (which may be in bit, byte, word, etc.) will temporarily suspend the sampling = disable. While a value different from "0" will enable the sampling.

**Run-Pause**

The name of the variable to be associated for setting the Trend's operating status is entered in this edit box (variable name can also be selected with the "..." browse button on the right). The operating status may be Run or Pause (Stop), unless the Trend has been configured to be managed in "Only Run" or "Only Pause": for which the variable has no influence.

The transition from "0" value to a different value of the associated variable (which may be in bit, byte, word etc.) changes the Trend's operating status. The variable will be put back to "0" by Movicon automatically.

- **Run Mode:** The Trend samples and represents the values of the current samples
- **Pause Mode:** The Trend continues to sample but displays historical data, by executing any queries in the associated Data Logger (if present) or if otherwise, displays the samples presently contained in the buffer

**Cursor In**

The name of the variable to be associated to the position where you wish to set the cursor for pointing sampled data displayed in Pause mode is entered in this box (or selected with the "..." browse button on the right). The cursor is represented by a vertical line in the centre of the Trend window.

The word type variable sets the desired number of samples to be pointed, with a value between 1 and the maximum number of samples value (buffer capacity) set.

**Cursor Out**

The name of the variable to be associated to the position obtained by cursor and returned by the Trend in Pause mode is entered in this edit box (variable name can also be selected with the "..." browse button on the right).

The cursor is represented by a vertical line in the centre of the Trend window.

The word type variable sets the desired number of samples to be pointed, with a value between 1 and the maximum number of samples value (buffer capacity) set.

**Scroll Next**

The name of the variable for executing the 'next value' command for the value pointed by the Trend's cursor in Pause mode is inserted in this edit box (variable name can also be selected with the "..." browse button on the right).

The change over from the associated variable's value from "0" to a different value (which may be in bit, byte word etc.) will bring forward the next sample, displaying a new value following the one pointed in the Trend's buffer.

**Scroll Prev.**

As for the "Scroll Next" variable, but for displaying a new value preceding the one pointed by the cursor in the Trend's buffer.

**Scroll Begin.**

The name of the variable for executing the command for displaying the first sampled value or contents in the Trend's buffer in Pause mode is entered in this edit box (variable name can also be selected with the "..." browse button on the right).

The change over of the associated variable's value from "0" to a different one (which may be in bit, byte word etc.) determines the pointing to the first value in the buffer and of its display in the Trend.

**Scroll End.**

As for the "Scroll Begin" variable, but for pointing and displaying the last sampling or value in the Trend's Buffer.



**Next Page**

The name of the variable for executing the "change page" command for the values displayed by the Trend in Pause mode is entered in this edit box (variable name can also be selected with the "..." browse button on the right).

The change over of the associated variable's value from "0" to a different one (which may be in bit, byte word etc.) determines the change page by displaying a new page of values following the value pointed in the Trend's Buffer.

**Prev. Page**

As for the "Next Page" variable, but for displaying a new page of the values preceding the value pointed in the Trend's buffer.

**Add Value**

The name of the variable for executing the recording of data on command is entered in this edit box (variable name can also be selected with the "..." browse button on the right). When this variable is set at "1" the Trend will execute a sampling and return the variable to "0". When the "Add Value" variable is inserted, the Trend will no longer record on time but on command only.

**Reset Values**

The name of the variable for executing the deleting of data in the Trend's buffer is entered in this edit box (variable name can also be selected with the "..." browse button on the right). When setting this variable at "1", the Trend will execute the deleting of data recorded up till that moment and return the variable to "0".

## Templates with Trends and Data Analysis

---

Templates are one of the most useful functionalities for the programmer to have to keep work safeguarded when working in a development environment. The Template philosophy is about allowing symbols from the Movicon library to keep their execution characteristics intact and which can also be applied to Trends and Data Analysis.

This argument has already been dealt with in the previous chapters relating to alarms, graphic animation, Data Loggers and the Movicon libraries and which we will also talk about here to include the Trends. These can receive rather articulated configurations and can also exploit this technology to the aim of being kept as templates in the symbols library under the Trends category.

The "Template" feature permits you to accelerate work on introducing and configuring the project's recording and displaying objects, permitting other "Database-Symbols-Trend" associations.

Let's suppose we need to realize a recording engine in a project which exploits the database technology and whose values have to be linked to a Trend object comprising of a chart symbol with trend related buttons and controls in order to carry out the selecting or custom analysis of data.

By using the Power Template technology we can realize the chart object with the Movicon graphics editor to build the Trends object and the command objects correlated to it, then configure the recording functions desired placed in relation to a database object inserted and configured in the Data Logger. The chart symbol can then be saved in the Symbols library and receive in association a DataLogger. It can then be inserted whenever needed, and Movicon will create the variables and databases related to the new Trend introduced into the screen and the project.

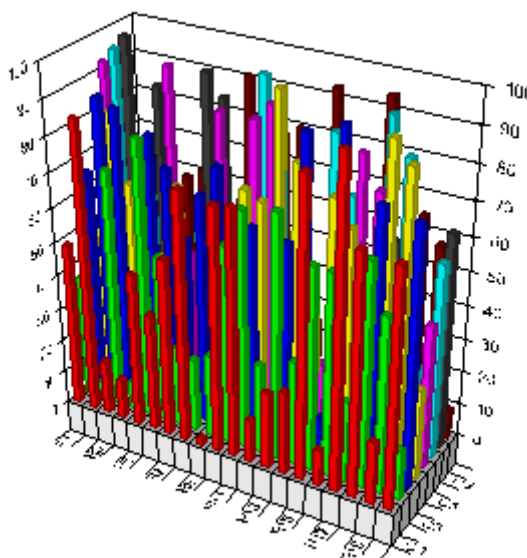
To apply the Template concept to Trends and the Data Loggers, you need to select the Trend object realized and saved in the Symbols library and use the **"Associate a DataLogger/Recipe"** command, which is accessed with the right mouse key, then select the DataLogger desired.

# Chart Objects

---

***Movicon has integrated a very powerful tool for representing project variables on two-dimensional and three-dimensional charts.***

Movicon has object windows which can be configured as pleased to create different kinds of charts in relation to the Movicon variables. These objects can be inserted in any Movicon screen by using the "chart" command from the "Trend-Charts-Analisi Dati". ToolBox. The charts are made up of objects have different style characters from the other Movicon objects seen so far which allow more powerful and sophisticated configurations.



**Charts use "Array" type variables. Therefore you need to be careful to the tips reporting the use of the variables in the charts.**

A Chart object can represent a series of historical curves, being retrieved data from a Data Logger, or dynamic curves which are data retrieved from a array of data. The curves which represent historical values retrieved from a Data Logger, must not be set with any variable and must have the same name as the Data Logger column where they are to be represented.



For further information on charts and their configurations, please consult the "First Impression 5.0 On-Line documentation" (VCFI5.HLP) online guide, available in the Movicon's installation folder.

## Chart Designer

---

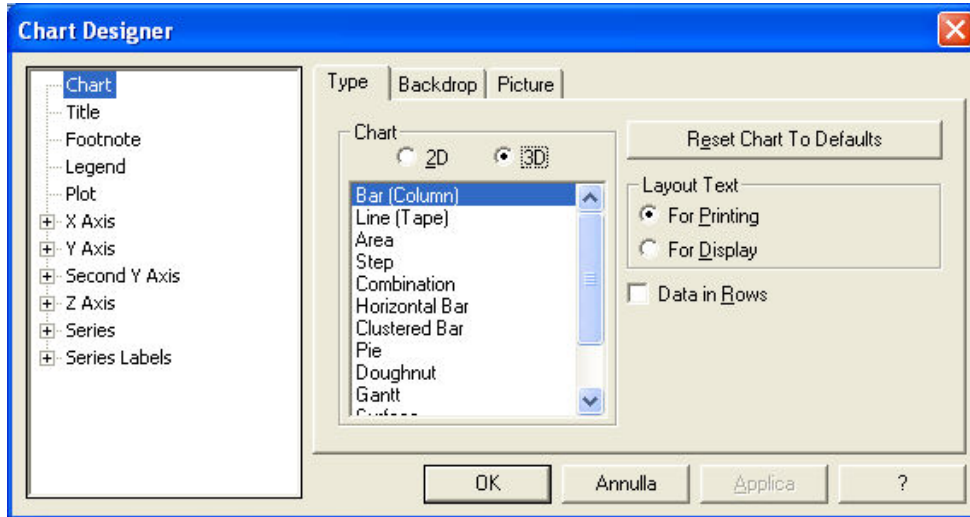
The chart's job is to represent value arrays of the associated variables when placed in a screen.

A chart can contain up to eight variables, configured as "byte Array" types of the same length.

- For example, if the VAR0001 variable is associated to a chart's Curve 1, previously inserted with a fixed length byte array (eg. with sizes of 10 bytes), you will get a 10 value chart representation (chart set with values in bytes). These

values will be represented individually by each single byte contents from the array type VAR0001 variable.

The style settings of the chart objects provide numerous possibilities. This chapter is aimed at describing the configuration applications most commonly used only, by taking into account two different types of charts. The user can use with the numerous style configuration properties to customize his/her chart as pleased.



## Configuring the Chart Styles

The charts' general styles and configurations can be setup through their setup windows.



The settings can be done in programming or in runtime mode (on condition that this has been provided by the programmer), by carrying out the very precise techniques as described below.

- During the programming phase, the chart's configuration is accessed through the "Open" button from the general properties window of the Chart properties.
- In Runtime mode, if enabled in the style property, the user can access the general style settings in two distinct ways:
  1. Right mouse clicking in the chart's proximity to access the setting command.
  2. Click to select the part of interest and then double-click.

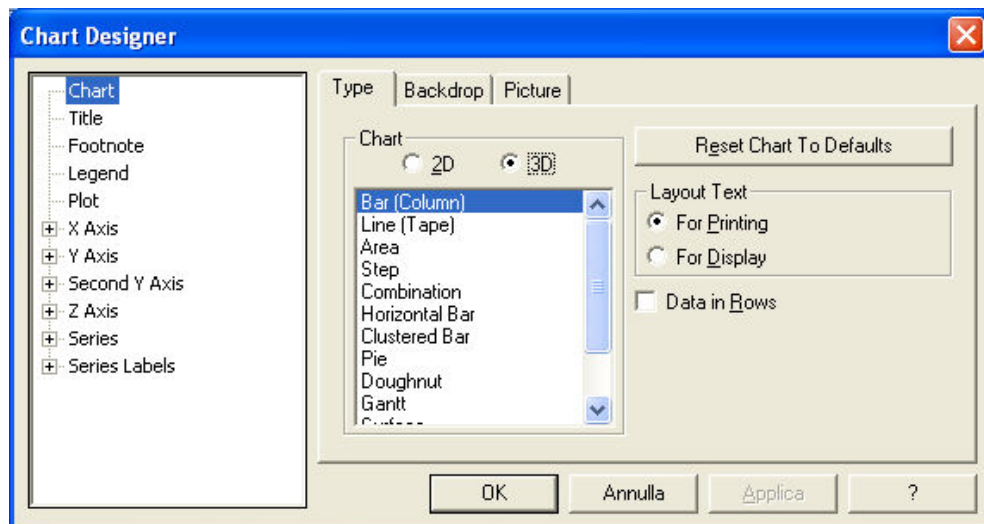
## General Chart Configurations

The general configurations consent to further settings to be done to the chart's style. Access to the General configurations can be obtained by activating the open button from the Chart's General property during the Movicon programming stage.



Apart from this, the General configurations can also be activated in runtime mode, if consented by the programmer, by using the right mouse key.

The general configurations provide numerous custom chart possibilities. In this chapter we will only describe those which are most important and commonly used.



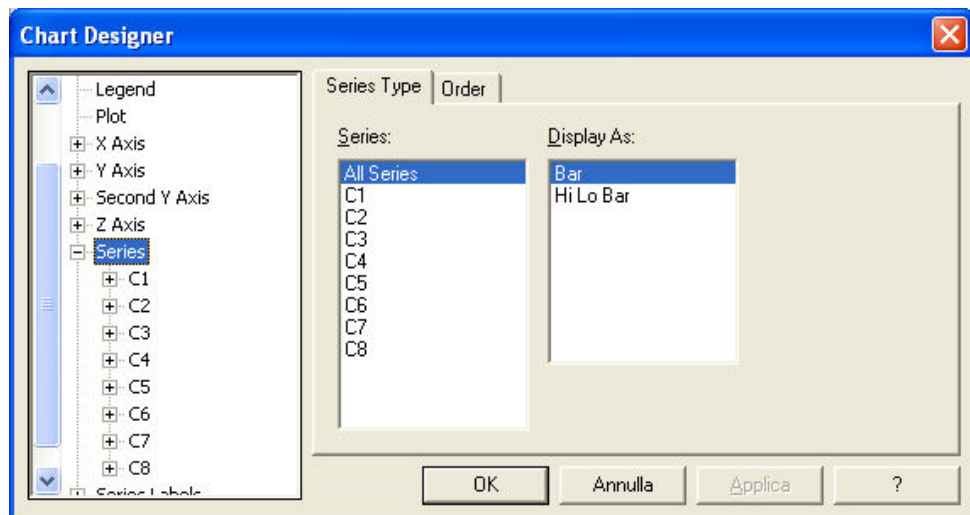
## Chart Editing

You can configure and customize the chart completely by using its general properties. The chart's design can also be adapted in runtime. To access to these features, some of which can only be edited by using the following modalities and using the mouse appropriately:

1. Click on the chart element to be edited (line, scale, grid, etc.). The element will appear highlighted with small squares
2. Double-click. The window for editing the settings will display

### Variable Format (series)

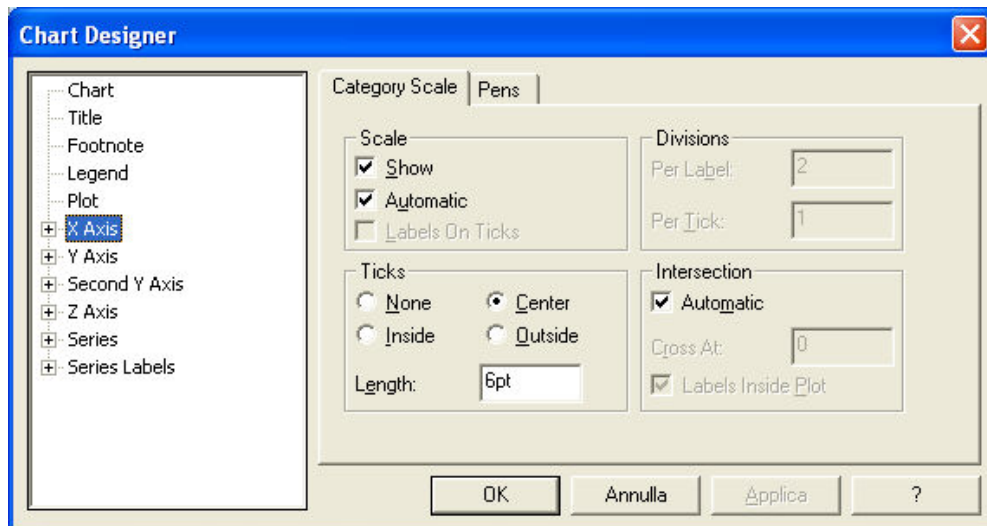
Pointing and double-click the chart's variables to activate the variable style settings window. You can edit the colors, sizes or styles of the variables representations in the chart by using these options.



### Axis format

Point and double-click the chart grid to activate the variable scale style setting window. These options allow you to change the color, size and style of the chart's variable scale. **This option is important for setting the two operating modes of the scale**

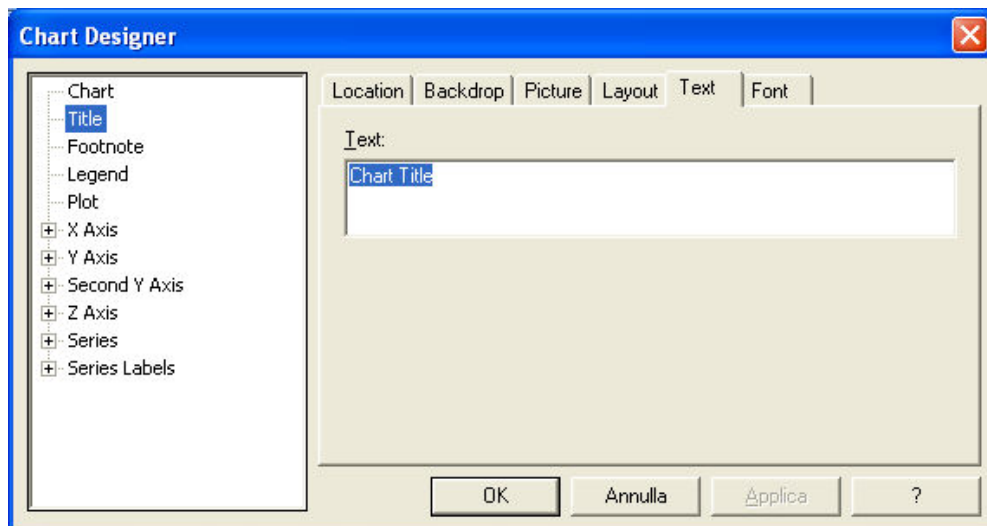
1. Automatic, with resizing of scale and chart as the variables change
2. Manual, with scale values whose sizes remain fixed as the variables change. In this case the sizes must be set manually.



### Axis Title Format

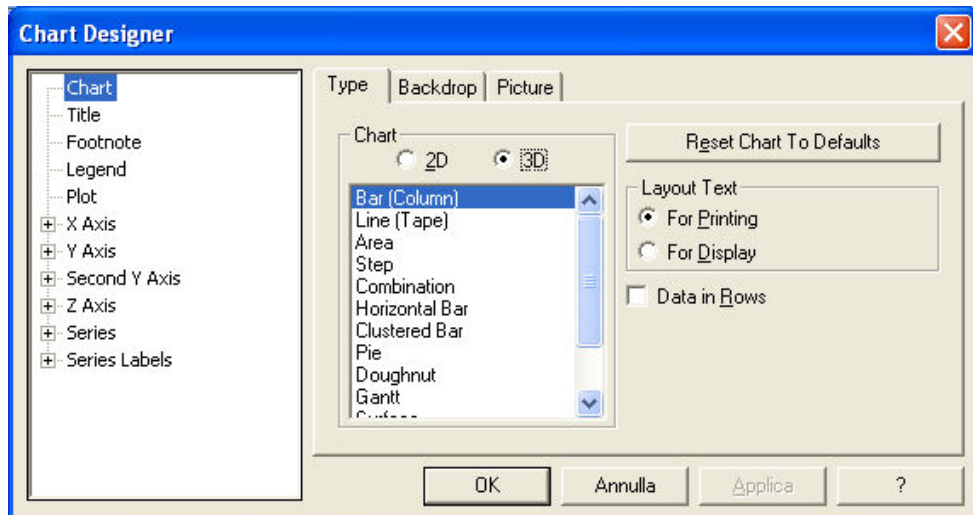
Point and double-click the titles at the side of the chart's axis scales to activate the window for the settings the axis titles.

You can edit the texts, fonts and colors of the titles displayed at the side of the axes in the chart.



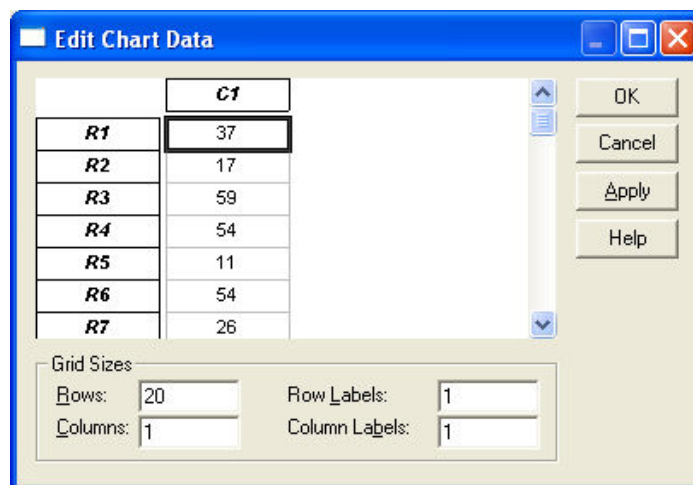
### Plot Format

Point and double-click the chart's background to activate the window for the chart's style settings, which can also be done through the general configurations.



## Editing Data and Labels

By using the "Edit Data and Labels", button found in the General Chart properties, you can access the window for setting the chart's label data during Runtime mode. This window allows the texts and default values associated to the chart labels to be edited. To edit a label or a value you need to select the data desired, overwrite the new text and confirm with OK.



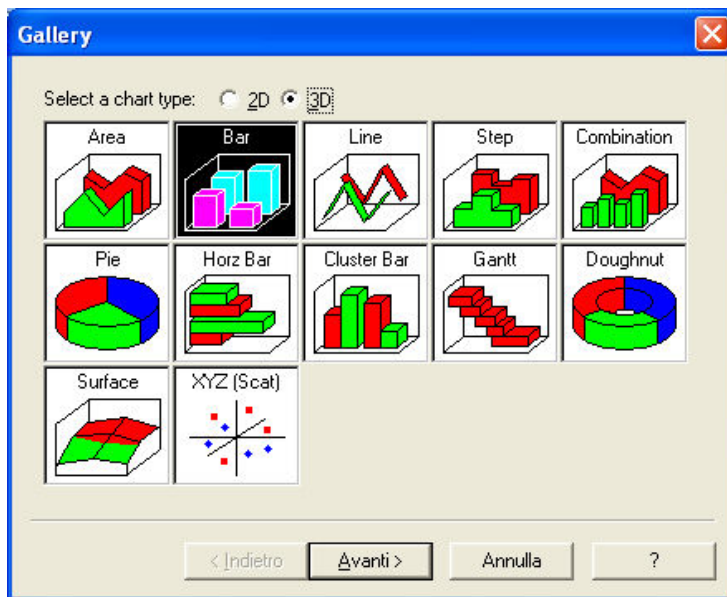
The data entered will replace the default labels displayed in the chart.

## Chart Wizard

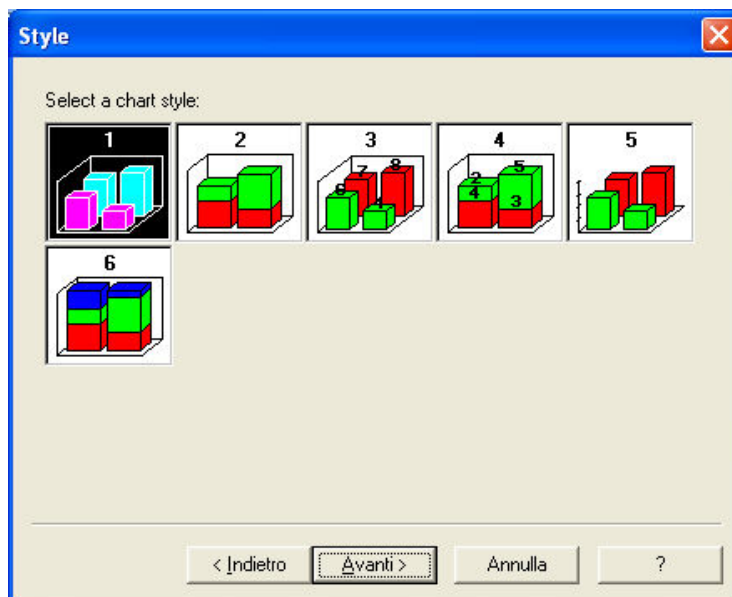
When entering a chart object on screen a Gallery window will appear with a simplified display of the style configurations with the use of a wizard to assist you in configuring the chart's style by carrying out four simple guided steps.

When having selected one of the options use the 'Next' button to continue on with the chart's settings which will display instantly.

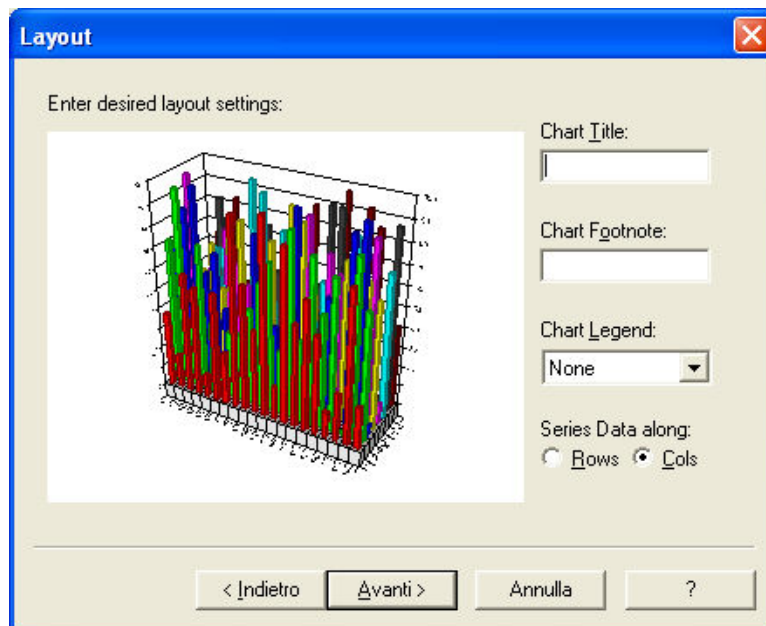
Select either the 2D (two-dimensional) or the 3D (three-dimensional) chart type option. Select the one of the icons representing the different chart aspects.



Note: remember that if you wish to have chart variables both on the X axis and Y axis, you need to select the 2D option and the XY chart type (or XYZ when choosing the 3D option). The associated variable arrays (max. 2/3, in this case), can be represented with their variations both in X and in Y (or in Z when in 3D).



The chart's style becomes easier and intuitive to set up through these type of configurations by directly selecting the chart desired from the corresponding images. By using the **Next** button you can continue on further with the style configuration by using the settings proposed.



Finally, a title can be assigned to the chart which will appear in the object according to any further settings which might be or may have been carried out in the General configurations.

The chart's layout can also be set to represent data either in Rows or in Columns.

## Chart Properties

The Chart objects which are inserted on screen can be completely customized in their properties. This can easily be done by selecting the Chart desired and then change its settings through the Movicon **"Properties Window"**.



The chart objects have two different types of properties. The Movicon functional properties, which can be set only in programming mode as for all the other objects, and the chart style properties which can be set through an appropriate window whether in programming or runtime mode.

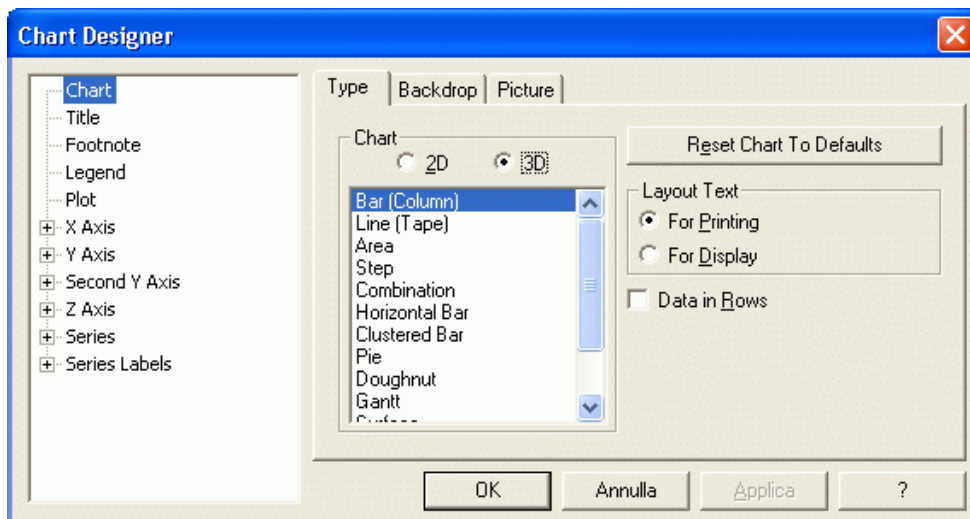
## Chart General Properties

As well as all the properties common to all the other Movicon drawings, you can also get access to the "Open" button for the chart's specific properties.

### Open

The "Open" button gives you access to the chart's configuration form. The configuration properties are documented with the relating Object's Help Online which can be activated with the Help button situated in each single form.





## Chart Execution Properties

By using the Execution properties of a Chart you can associate it with variables and modalities of functioning. In order to do this, select the Chart object on the screen and then change its settings through the Movicon **"Properties Window"**.

### Curve Variables

The chart can be managed with the maximum of 8 graphically represented values. These values have to be associated through the project's variables by using the execution properties' Curve Var. <n> boxes. The names of the variables you wish to insert into the chart have to be written in these boxes by selecting them from those existing in the Real Time DataBase.

You must keep in mind that the curves represent arrays of data therefore you must take note of the array length specified in the Num. Samples box when addressing the successive curves.



**Important: The Charts use exclusively "Array" type variables for data for which the associated variables have to be configured appropriately in their properties as "Fixed Length Byte Array ". Remember that the array's "Fixed Length" must always be specified in the variable's properties in the "Address" box by setting the number of bytes desired between the brackets after the address.**



When a XY chart is being used the two variable arrays indicate the scale for the chart's X axis and for the Y axis. This concept will be further dealt with ahead.

### Rotation Variable

A variable can be set in this property for rotating the chart three-dimensionally. The visual rotation angle of the 3D chart will then be influenced by the value of the associated value in runtime.

### Elevation Variable

A variable can be set in this property to rotate the 3D chart vertically. The visual rotation angle of the 3D chart will then be influenced by the value of the associated value in runtime.

### Push Value Variable

The current Chart displays all the values of the assigned array variables. A "Push" variable is being designed for updating the chart's values upon the status change of that variable and will be available in the near future.

**Reset Value Variable**

A variable can be specified for executing the reset of the Chart's data. The command will activate when this variable is set at a value different from zero, after which it will be reset to zero by Movicon.

**Num. Samples**

This edit box is used for setting the number of values (samples) to be represented on the chart. The default value (20) means that the chart displays 20 values in function with the data type set, independently of the size of the array which is expressed in bytes and refers to the variable.

**Editable**

When this selection is enabled, the chart will become editable for the operator during Runtime. This also means that the chart's assigned style settings can be changed in its style properties during runtime.

**Array Type**

This edit box is used to indicate to the chart how to use the values of the associated Array variables. The Array variables are always expressed in bytes. When the value contents are to be represented on the chart in word, dword or other, you need to select the data type desired, independently of the array's unit measures in bytes.

**Network Server**

This edit box allows you to specify the name of any eventual Network Server from where data is to be retrieved. In this way the Chart control will be able to ask the Movicon server when retrieving data instead of getting data from the local Data Logger's DBMS link to the project. This allows historical data from a remote Data Logger to be displayed on a remote PC in chart format.

To get the Chart control to retrieve historical data from a remote Data Logger, you need to set the following properties in the Chart object:

- Execution -> Network Server
- Execution -> Max. Rows
- Style -> Data Logger Link

In addition to this, the Data Logger linked to the Chart must also be configured in the local project. However, you can disable its 'Enable' property to avoid that recordings on Database, also local, are carried out.

**Network Backup Server**

See paragraph "Drawings and Controls common Execution Properties".

**Max Rows**

This edit box allows you to insert the maximum number of records to be retrieved from the Server's Data Logger.

## Chart Style Properties

---

The Chart Style properties allow name associations to the values of the chart.

Therefore you can replace the default name by specifying another name which will be displayed in the title of the chart's value.

**Curve Title**

This property allows you to associate a name to the chart's curves.

**Data Logger Linked**

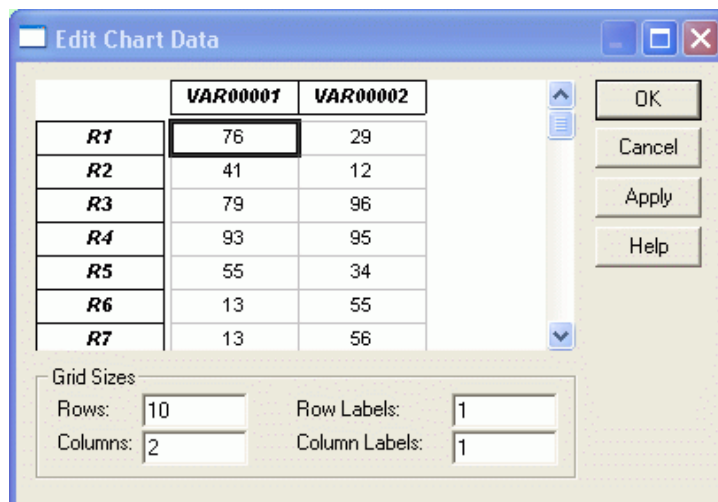
This property allows you to specify the name of the Data Logger to be associated to the Chart.

**Refresh Pens from Data Logger Linked**

This button allows the curves to be refreshed according to the linked Data Logger's structure. This command simply sets the "Curve Name" property with the same name of the linked Data Logger's columns.

### Edit Chart Data

This button allows you to open a dialog window through which you can edit the labels which identify the chart's rows and column:



	VAR00001	VAR00002
R1	76	29
R2	41	12
R3	79	96
R4	93	95
R5	55	34
R6	13	55
R7	13	56

Grid Sizes

Rows: 10      Row Labels: 1

Columns: 2      Column Labels: 1

### Border

The **"Border"** is described in the paragraph titled **"Style Proprieties common to Drawings and Controls"**.

## DataLogger Window

The Movicon **'DataLogger Window'** is a configurable object designed to display recorded data from the Project's DataLoggers or Recipes. Data from local projects and remote stations (Server) connected in net can also be displayed.

The DataLogger Window object is available from the 'Special Objects' group in the **"Objects Window"**.

When a DataLogger Window is inserted into a screen it will appear with its default size. After being inserted it can be resized as pleased by dragging the borders with the mouse. Each DataLogger Window can display data of only one DataLogger or Recipe which must be setup in the object's **"Execution Properties"**.



As many DataLogger Window in as many screens can be inserted within the project according to your specific requirements.



The DataLogger or Recipe associated to the DataLogger Window can be dynamically changed during Runtime by using the object's appropriated Basic Script functions.

## DataLogger Window Fields

The fields which must be presented in the DataLogger Window can be selected by the programmer from those available by means of using the **"Field Choice"** window.

The fields or columns which can be displayed are described below:

### Event Time

The 'Event Time' column reports the date and time of the data recording. The time is always in the operating system's local time and will be displayed with the "YYYY-MM-GG HH-MM-SS" formation.

**Reason**

The 'Reason' column reports the cause which determined the recording. Recordings can take place on time, on change or on variable.

**User**

The 'User' column reports the name of the user who was active when the recording took place.

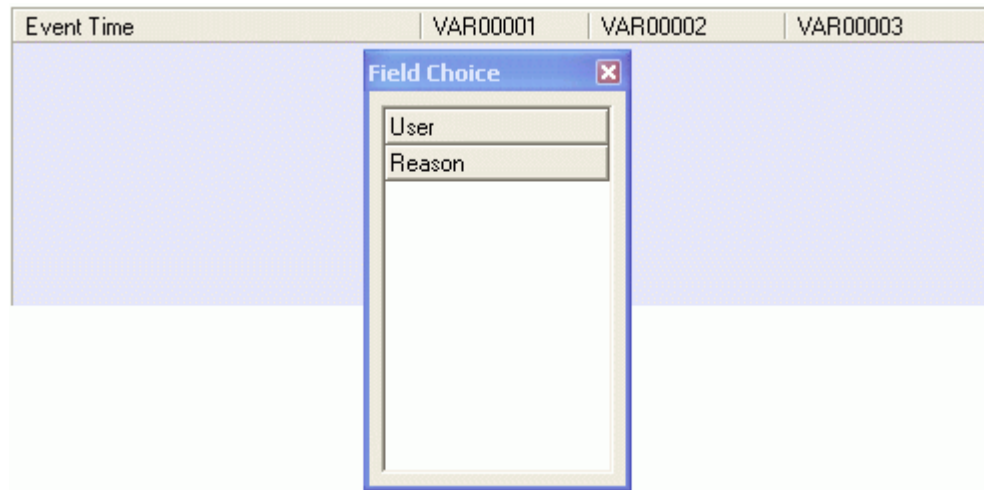
**Variable**

A column is created in the database for each variable associated to the DataLogger or Recipe with the same name set in the "**Data Loggers and Recipes common Column General Properties**".

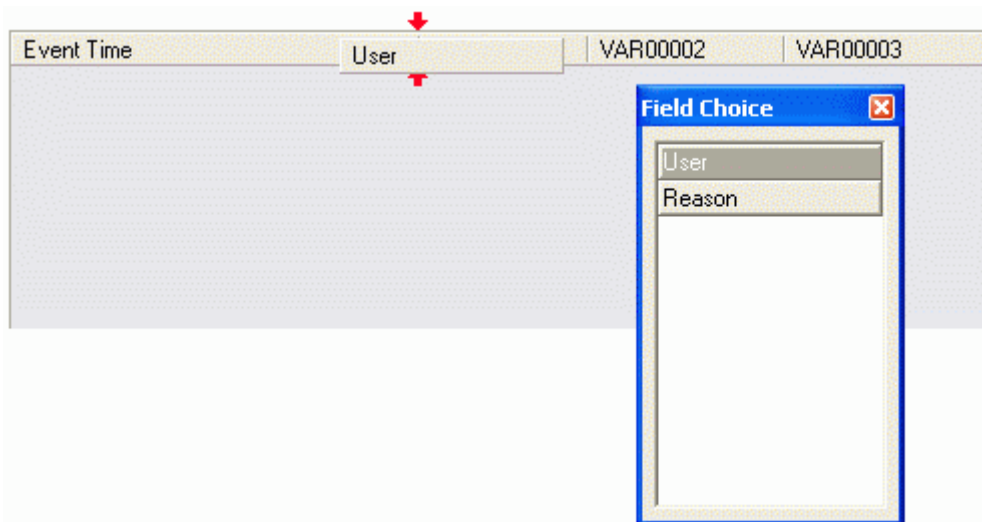
## DataLogger Window Field Choice

---

The DataLogger Window object permits you to select the fields and columns to be displayed. In order to do this you need to open the '**Field Choice**' window by clicking on the '**Open**' button from the '**General**' group in the object's '**Properties Window**' or by double clicking and keeping the mouse button pressed on the object while pressing the '**SHIFT**' key at the same time.



The available fields which have not yet been inserted into the DataLogger Window will be listed in the 'Field Choice' window. To move a field from the 'Field Choice' window to the DataLogger Window just select the field with the mouse and drag it to the position desired in the DataLogger Window. Note that the field can only be released at the beginning or end of an already positioned field, and that two red arrows should appear:



To move a field from the DataLogger Window to the 'Field Choice' window just select the field with the mouse and drag it into the 'Field Choice' window.



When the window has been set with the "Show Control Wnd" option, the columns can be dragged out of the window and deleted during Runtime mode. However, the initial configuration can be restored by using the "RecalcLayout" basic script method from the DLRWndCmdTarget interface when the page is reloaded.

## DataLogger Window Style Properties

The Style properties of the DataLogger Window are used for setting the object's graphical properties.

To modify the Style properties, select the object with the mouse and use the Movicon **'Properties Window'**.

### Border

The **'Border'** property is described in the paragraph on **"Style Properties common to Drawings and Controls"**.

### Clickable

The **'Clickable'** property is described in the paragraph on **"Style Properties common to Drawings and Controls"**.

### Auto Layout

The **'Auto Layout'** property is described in the paragraph on **"Style Properties common to Drawings and Controls"**.

### Show Control Wnd

The **'Show Control Wnd'** property is described in the paragraph on **"Style Properties common to Drawings and Controls"**.

### Refresh Button

The **'Refresh Button'** property is described in the paragraph on **"Style Properties common to Drawings and Controls"**.

### Filter Button

The **'Filter Button'** property is described in the paragraph on **"Style Properties common to Drawings and Controls"**.

**Print Button**

The '**Print Button**' property is described in the paragraph on "**Style Properties common to Drawings and Controls**".

**Button Size**

The '**Button Size**' property is described in the paragraph on "**Style Properties common to Drawings and Controls**".

**Align Buttons**

The '**Align Buttons**' property is described in the paragraph on "**Style Properties common to Drawings and Controls**".

**Refresh Button Text**

The '**Refresh Button Text**' property is described in the paragraph on "**Style Properties common to Drawings and Controls**".

**Filter Button Text**

The '**Filter Button Text**' property is described in the paragraph on "**Style Properties common to Drawings and Controls**".

**Print Button Text**

The '**Print Button Text**' property is described in the paragraph on "**Style Properties common to Drawings and Controls**".

**Time Column Name**

The text which will appear as the "**Event Time**" column's name is entered in this edit box. The default text will be used when this field is left empty.

**User Column Name**

The text which will appear as the "**User**" column's name is entered in this edit box. The default text will be used when this field is left empty.

**Reason Column Name**

The text which will appear as the "**Action**" column's name is entered in this edit box. The default text will be used when this field is left empty.

**Include milliseconds**

The '**Include milliseconds**' property is described in the paragraph on "**Style Properties common to Drawings and Controls**".

## DataLogger Window Execution Properties

---

The Execution properties of DataLogger Window are used for setting the DataLogger or Recipe to be displayed and the project it belongs to. To modify the Execution property, select the object with the mouse and use the Movicon '**Properties Window**'.

**Project**

The name of the child project from where data is to be taken is entered in this edit box. If this field is left empty the current project will be used.



Only the name of an eventual child project of the current project is entered in this property.

**Data Logger-Recipe**

The name of the DataLogger or Recipe to be displayed is to be entered in this edit box. The '...' browse button on the far right allows you to select the one needed from the project's list of 'DataLoggers & Recipes'. When the DataLogger or Recipe is not on the list because it belongs to another project, just simply enter the name using the keyboard.

**Max. Rows**

The highest number of rows which can be displayed in the DataLogger Window is entered in this edit box.

### Network Server

See paragraph "Drawings and Controls common Execution Properties".



In order to display data from a Server it is necessary that the DataLogger/Recipe be also present in the Client project, to recuperate the Database structure. The DataLogger/Recipe can be created as a structure only and therefore without having to associate any variables to the columns.

### Network Backup Server

See paragraph "Drawings and Controls common Execution Properties".

## DataLogger Data Filter

Filters on displaying data in the DataLogger window can be executed during a project runtime mode. These filters can be done by following different data selecting criteria. When the **"Filter"** button, in the DataLogger Window, is pressed the following window will open for setting the filter:



When applying more than one filter at the same time you have to put "AND" between each one in the extraction query, then the conditions will be added up together.



You can customize the font and size of the Dialog window's characters by using the right registry key:

DBFilterFont  
DBFilterFontSize

The size of the dialog window will also change when the font or its size is modified.

### Sort by:

This selection is used for putting data into order according to the DataBase column chosen. The data order can be done by simply clicking on the column desired in the DataLogger Window.

### User:

This entry field is used for specifying the name of the user who wants to execute the filter. By doing this, only the data recorded while the user was logged on will be extracted.



The text to be entered in this field is not the name of the user but their "Description or Electronic Signature".

**From Date:**

This field is used for entering the date with which the data extraction is to start from.

**To Date:**

This field is used for entering the date which the data extraction is to end with.

## TraceDB Window

---

The Movicon **'TraceDB Window'** is a configurable object designed to display variable trace data. Not only can data from local projects be displayed but also data from remote stations (Server) connected in net.

The 'TraceDB Window' object is available from the 'Special Objects' group in the **"Objects Window"**.

The TraceDB Window will appear with its default settings when inserted into a screen. After it has been inserted, it can be resized as pleased by dragging its borders with the mouse.

Each TraceDB Window can display the data one variable only which must be set in the object's **"Execution Properties"**.



As many TraceDB Window in as many screens can be inserted in the project as required.



The variable associated to the TraceDB Window can be dynamically changed during Runtime by using the object's appropriated Basic Script functions.

## TraceDB Window Fields

---

The fields which must be presented in the TraceDB Window can be selected by the programmer from those available by means of using the **"Field Choice"** window.

The fields or columns which can be displayed are described below:

**Event Time**

The 'Event Time' column reports the date and time of the data recording. The time is always in the operating system's local time and will be displayed with the "YYYY-MM-GG HH-MM-SS" formation.

**User**

The 'User' column reports the name of the user who was active when the recording took place.

**Action**

The 'Action' column reports the reason causing the variable variation. For instance the variation may derive from a screen control, (display, button, etc), the communication driver, or from the Movicon Watch window, etc.

**Value**

The 'Value' column reports the requested value to be set on the variable. In certain cases there might not be the conditions needed to activate the value requested and therefore the final value may not be the one requested.

**Before**

The 'Before' column reports the value of the variable before changing.



### After

The 'After' column reports the final value obtained after being changed. In certain cases this value might not correspond to the one requested, for instance if the writing is not executed properly.

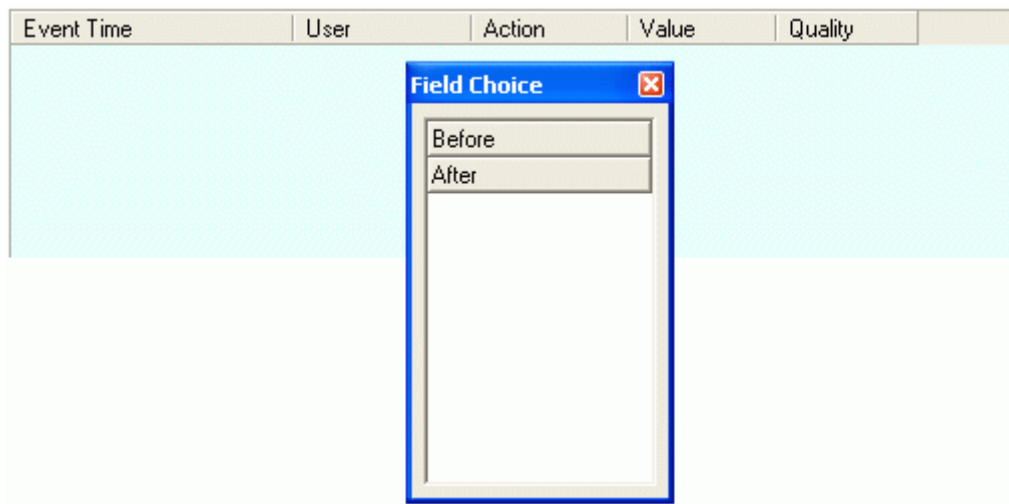
### Quality

The 'Quality' column reports the quality of the variable at the time of the recording.

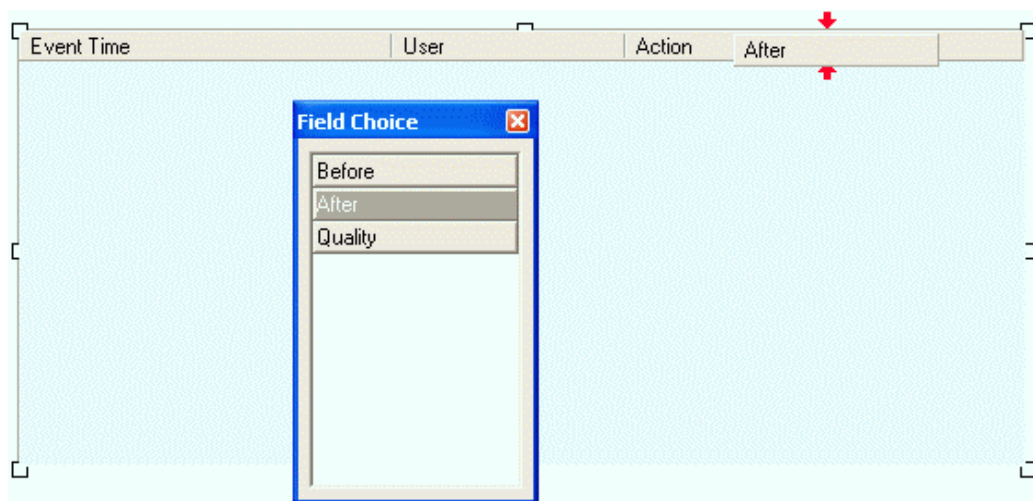
## TraceDB Window Field Choice

The TraceDB Window object permits you to select the fields or columns to be displayed.

In order to do this you need to open the '**Field Choice**' window by clicking on the '**Open**' button from the 'General' group in the object's '**Properties Window**' or by double clicking and keeping the mouse button pressed on the object while pressing the '**SHIFT**' key at the same time.



The available fields which have not yet been inserted into the TraceDB Window will be listed in the 'Field Choice' window. To move a field from the 'Field Choice' window to the TraceDB Window just select the field with the mouse and drag it to the position desired in the TraceDB Window. Note that the field can only be released at the beginning or end of an already positioned field, and that two red arrows should appear:



To move a field from the TraceDB Window to the 'Field Choice' window just select the field with the mouse and drag it into the 'Field Choice' window.



When the window is set with the "Show Control Wnd" option, the columns can be dragged out of the window during Runtime where they will be deleted. You can, however, restore the initial configuration by using the "RecalcLayout" basic script method from the TraceDBWndCmdTarget interface or when the page is reloaded.

## TraceDB Window Style Properties

---

The style properties of the TraceDB Window are used for setting the object's graphical properties.

To modify the Style properties, select the object with the mouse and use the Movicon **'Properties Window'**.

### **Border**

The **'Border'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

### **Clickable**

The **'Clickable'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

### **Auto Layout**

The **'Auto Layout'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

### **Show Control Wnd**

The **'Show Control Wnd'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

### **Refresh Button**

The **'Refresh Button'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

### **Filter Button**

The **'Filter Button'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

### **Print Button**

The **'Print Button'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

### **Report File**

This property's description can be found in the paragraph on Drawing and Control's "Style Objects Proprieties".

### **Edit Report File**

This property's description can be found in the paragraph on Drawing and Control's "Style Objects Proprieties".

### **Button Size**

The **'Button Size'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

### **Align Buttons**

The **'Align Buttons'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

### **Refresh Button Text**

The **'Refresh Button Text'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

### **Filter Button Text**

The **'Filter Button Text'** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

**Print Button Text**

The '**Print Button Text**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

**Time Column Name**

The text which will appear as the "**Event Time**" column's name is entered in this edit box. The default text will be used when this field is left empty.

**User Column Name**

The text which will appear as the "**User**" column's name is entered in this edit box. The default text will be used when this field is left empty.

**Action Column Name**

The text which will appear as the "**Action**" column's name is entered in this edit box. The default text will be used when this field is left empty.

**Before Column Name**

The text which will appear as the "**Before**" column's name is entered in this edit box. The default text will be used when this field is left empty.

**After Column Name**

The text which will appear as the "**After**" column's name is entered in this edit box. The default text will be used when this field is left empty.

**Value Column Name**

The text which will appear as the "**Value**" column's name is entered in this edit box. The default text will be used when this field is left empty.

**Quality Column Name**

The text which will appear as the "**Quality**" column's name is entered in this edit box. The default text will be used when this field is left empty.

**Include milliseconds**

The '**Include milliseconds**' property is described in the paragraph on "**Style Proprieties common to Drawings and Controls**".

## TraceDB Window Execution Properties

---

The Execution properties of the TraceDB Window are used for setting the table to be displayed and what project it belongs to.

To modify the Execution properties, select the object with the mouse and use the Movicon '**Property Window**'.

**Project**

The name of the child project from which variable's trace data is to be retrieved is entered in this edit box.

If this field is left empty the current project will be used.



Only the name of an eventual child project of the current project is entered in this property.

**Variable**

The name of the trace table to be displayed is to be entered in this edit box. The '...' browse button on the far right allows you to select one of the Real Time DB variables. This method is very handy when the name of the trace table is the same as the variable's. However if the table's name has been customized through the "**Table Name**" settings of the '**Variable Trace Option Properties**', it will have to be entered by using the keyboard.



The 'Variable Table' property requires the name of the Trace Database table and not the name of the associated Variable.

**Max. Rows**

The highest number of rows that can be displayed in the TraceDB Window is entered in this edit box.

**Network Server**

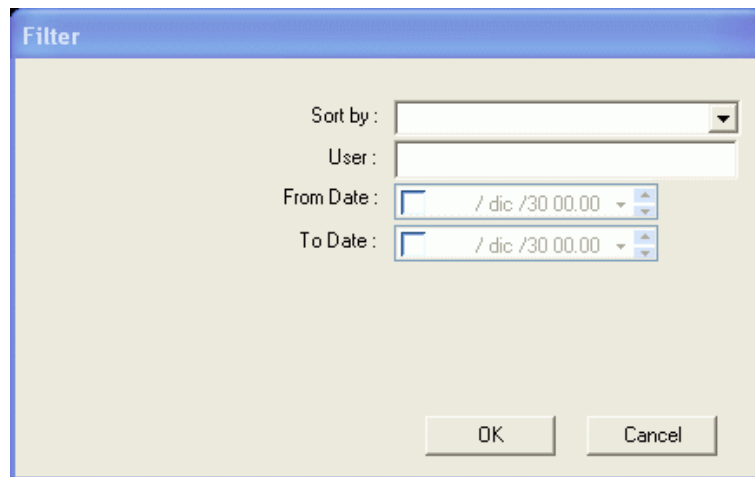
See paragraph "Drawings and Controls common Execution Properties".

**Network Backup Server**

See paragraph "Drawings and Controls common Execution Properties".

## TraceDB Data Filter

Filters for displaying data in the Trace Log window can be executed during a project runtime mode. These filters can be done by following different data selecting criteria. When the **"Filter"** button in the Trace Window is pressed, the following window will open for setting the filter:



When applying more than one filter at the same time you have to put "AND" between each one in the extraction query, then the conditions will be added up together.



You can customize the font and size of the dialog window characters by using the appropriate registry keys:

DBFilterFont  
DBFilterFontSize

The dialog window sizes will change when you modify the font and its size.

**Sort by:**

This selection is used for putting data into order according to the DataBase column chosen. The data order can be done by simply clicking on the column desired in the Trace Window.

**User:**

This entry field is used for specifying the name of the user who wants to execute the filter. By entering the user's name, only data recorded during their log on will be extracted.



The text to be entered in this field is not the "Name" of the user, but their "Description or Electronic Signature".

**From Date:**

The date with which the extractions are to start from is entered here.

**To Date:**

The date with which the extractions are to finish is entered here.

# The Grid

***The Grid object allows table representations of data archived on DataBase files or text files.***

The Grid object allows the data contents of DataBase tables to be displayed in table format. The data to be displayed can be part or totally extracted according to the set query.

The Grid object belongs to the Movicon "Advanced Shapes" class and therefore can be inserted on screen by using the **"Objects Window"**.



The Grid is a powerful display tool of data in DataBase format which also allows values to be changed on DataBase Tables.

	VAR00001	VAR00002	VAR00003
1	45	44	5
2	45	6	5
3	34	6	5
4	34	6	5
5	34	6	5
6	34	6	5
7	12	33	5
8	56	43	43

Update (Ctrl+U)

Save (Ctrl+S)

Select All (Ctrl+A)

Copy (Ctrl+C)

Insert (Ins)

Delete (Del)

This is an example of a Grid displaying the contents of a DataBase.

When the Grid's "Clickable" style property is set you can edit the Grid boxes and save changes on DataBase.

In addition to the files in Database format, the Grid is also capable of displaying the data contained in text files (UNICODE format) providing that the data is separated by a preset character. The comma (",") is the character used for default a different one can be setup through the appropriate properties.

## The Grid Buttons

The Grid object allows you to execute some commands during the Runtime phase by using the buttons described below.

### Refresh (F5)

This command updates the grid by reloading data from the linked file.

### Update (Ctrl+U)

This command updates the variables which have the same name of the table columns with the record values of the grid selected.

### Save (Ctrl+S)

This command saves the table's data on DataBase. For example, any field changes are recorded on DataBase files.

**Select All (Ctrl+A)**

This command selects all the table's data.

**Copy (Ctrl+C)**

This command copies of all the data selected in the table onto the Windows clipboard.

**Insert (Ins)**

This command inserts a new line in the table.

**Delete (Del)**

This command deletes the line or lines selected in the table.

## Grid Style Properties

---

The Grid Style properties are used for setting up the object's graphic properties.

To edit the Style properties, select the object with the mouse and use the Movicon **"Properties Window"**.

**Border**

The **"Border"** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

**Clickable**

The **"Clickable"** is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

**Show Control Wnd**

The **"Show Control Wnd"** is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

**Auto Layout**

The **"Auto Layout"** is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

**Refresh Button**

This property enables or disables the displaying of the Refresh button.

**Update Button**

This property enables or disables the displaying of the Update button.

**Save Button**

This property enables or disables the displaying of the Save button.

**Select All Button**

This property enables or disables the displaying of the Select All button.

**Copy Button**

This property enables or disables the displaying of the Copy button.

**Insert Button**

This property enables or disables the displaying of the Insert button.

**Delete Button**

This property enables or disables the displaying of the Delete button.

**Button Size**

The **"Button Size"** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

**Align Buttons**

The **"Align Buttons"** property is described in the paragraph on **"Style Proprieties common to Drawings and Controls"**.

**Refresh Button Text**

This edit box allows you to insert the text to be associated to the Refresh button. When this field is left empty the default text will be used instead.

**Update Button Text**

This edit box allows you to insert the text to be associated to the Update button. When this field is left empty the default text will be used instead.

**Save Button Text**

This edit box allows you to insert the text to be associated to the Save Button. When this field is left empty the default text will be used instead.

**Select All Button Text**

This edit box allows you to insert the text to be associated to the Select All Button. When this field is left empty the default text will be used instead.

**Copy Button Text**

This edit box allows you to insert the text to be associated to the Copy Button. When this field is left empty the default text will be used instead.

**Insert Button Text**

This edit box allows you to insert the text to be associated to the Insert Button. When this field is left empty the default text will be used instead.

**Delete Button Text**

This edit box allows you to insert the text to be associated to the Delete Button. When this field is left empty the default text will be used instead.

**Update Variable**

This property allows you to enable the possibility to update the variables which have the same names as the Table columns with the record values of the selected grid. Updating is done on the Update button's command.

## Grid Execution Properties

---

The Grid execution properties are used for setting up file connections for display data. To edit the Execution properties, select the object with the mouse and use the Movicon **"Properties Window"**.

**ODBC DSN**

This property allows you to insert or create the ODBC DSN connection to be used for accessing the Database to be displayed in the Grid.



The ODBC is not supported in WinCE for the time being and therefore the Database file DSN link cannot be exploited and the Grid's "ODBC DSN" property has to be changed during its configuration phase. This setting will then be different according to whether the Grid has to be linked to a DataLogger/Recipe or to any Database file. In the first case, Movicon will have already opened connections to the file and therefore the Grid should be able to exploit them. In the second case, however, the Grid will have to open a new connection to the DataBase file. As a consequence of this, the "ODBC DSN" property should be set as follows:

- **Connection to DataLogger/Recipe:** the DSN name in the "ODBC DSN" property is replaced with the name of the DataLogger/Recipe
- **Connection to DataBase:** The DSN name in the "ODBC DSN" property is replaced with the DataBase file name to be connected to, without specifying the extension. The DataBase file, in this case, must be found in the project's DLOGGERS folder or one of its sub folders

**ODBC DSN User**

This property specifies the users name to be used for the ODBC connection.

**Query**

The extraction query of data from the selected database table is inserted in this edit box. Only data extracted according to the query's parameters is displayed in the Grid.

**Text FileName**

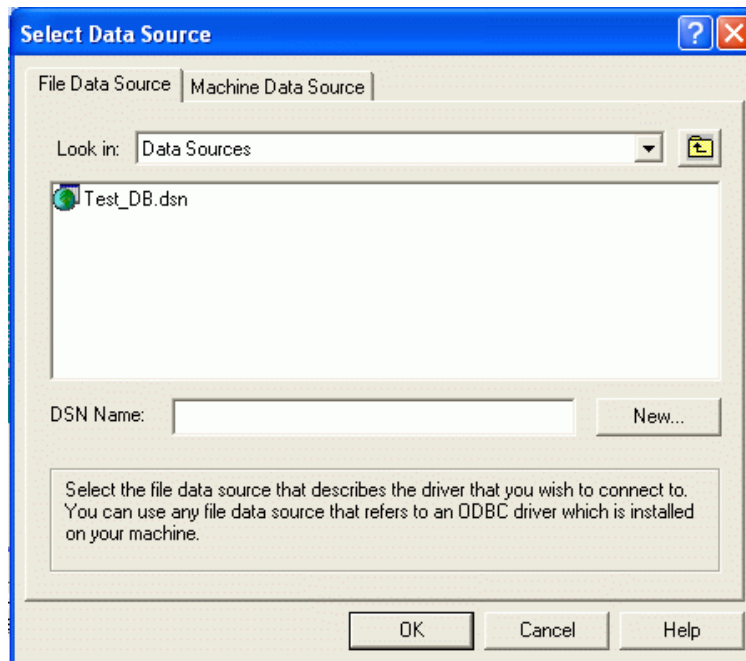
This property allows you to insert or select the name of the text file to be displayed in the Grid instead of a Database file.

**Text Separator**

This property allows you to set the separators to be use for separating data in the selected text file to be displayed in the Grid. When this field is left empty the "," default separator character will be used.

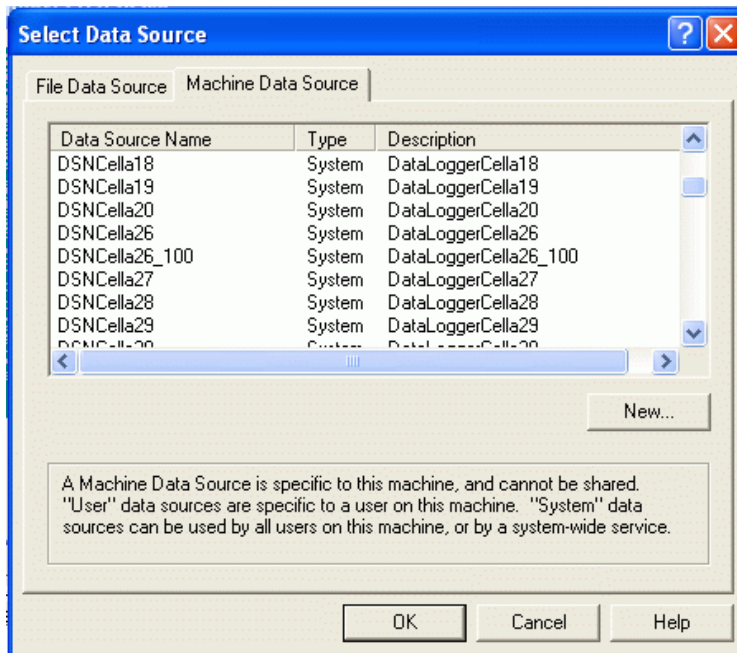
## Data Source Selection (DSN)

The object Grid allows a Database file to be displayed by exploiting its ODBC DSN connection. It is necessary that there be a DSN Data source for the Database file which you wish to display. To select the data source to be associated to the Grid object you can use the "Open" command from the object's General properties or double click on the object while keeping the SHIFT key pressed down. The following window will appear:

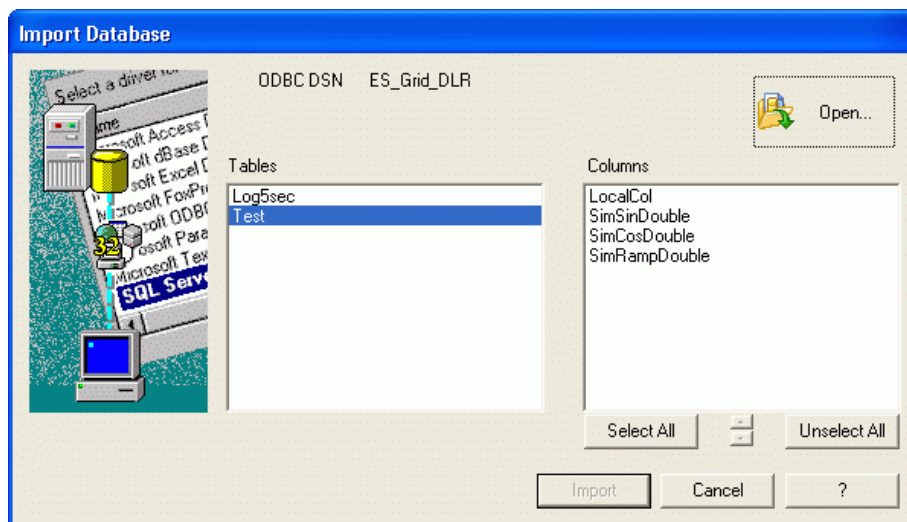


You can select or create a file data source by using the "New..." from this window. If you select the other tab you will be able to select the machine data source:





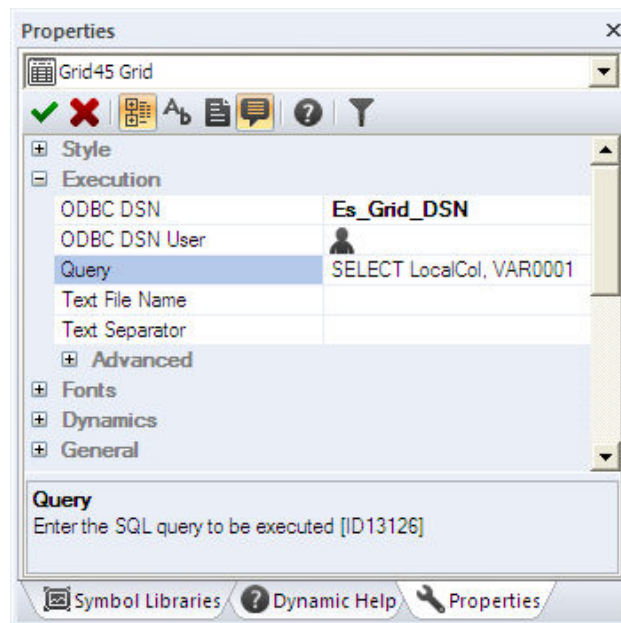
Once the data source has been selected a window will appear with a list of tables available in the Database, and by selecting the table of interest you will get a list of its fields on the right hand side.



You can select or deselect the table's fields to import the table columns need. A SELECT query will be generated based on the selection made to extract the data to be displayed on the Grid.

The "Open" button allows you to select a new data source, by re-showing the previous window.

The "Import" button ends the data source selection procedure by creating a select query which will then be returned to the **"Query"** property of the **"Grid Execution Properties"** group.



The recognized order in which the data is extracted is the one set by the select query. If the query is changed manually there may no longer be any coherence between the Grid's column titles and the data displayed in those columns. In order to change the Grid's structure as pleased it would be best to use the Basic Script functions for inserting and putting the columns into the order your desire. If you wish to change the columns during the programming phase you can interact on the object's XML codes as usual.

# Hour Selector

The Movicon **"Hour Selector"** is a configurable object which is associated to an object from "Scheduler Object List" resource to be used for viewing and changing the Scheduler object's daily plans.

The "Hour Selector " object is found in the "Advanced Objects" group from the **"Objects Window"**.

When an "Hour Selector" is inserted on screen you will be able to view and change the times set in the Scheduler object's hour plan, it is associated to, in Runtime.



A "Hour Selector" can also display the daily plan of a Scheduler setup in a Network Server project by using the appropriate property.

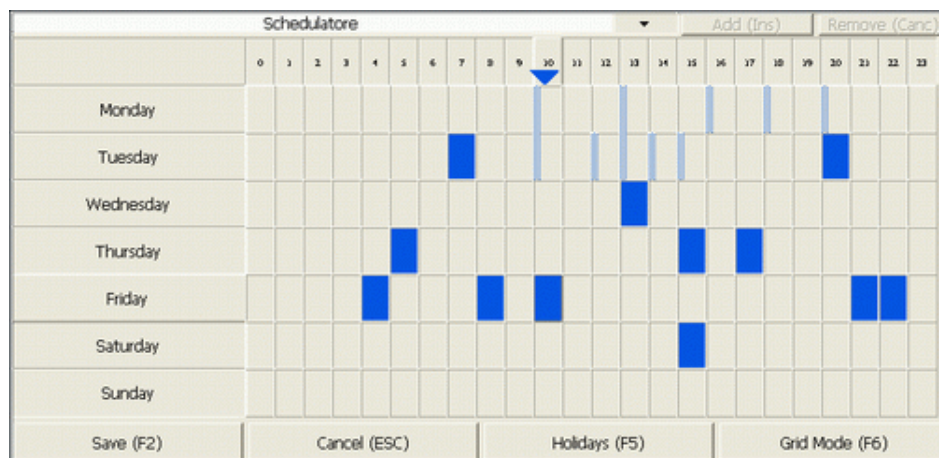
The "Scheduler Window" can manage activation hour entries in two different modes:

- **Time Table**
- **Grid**

Either one or both can be set for use in the designing phase. Keep into account that once a change has been made in the "Grid" mode, you will not be able to change the entered data in the "Time Table" mode.

## Time Table

This type of display allows you to graphically view the times preset in the Scheduler selected using the list-box at the top. In this way you can modify the times set by clicking the mouse on the box, as described above, in 15 minute resolutions. Only activation times of commands can be changed and not the variables associated to them. The "scheduler Window" displays as below:



A Scheduler Window displayed in 'Time Table' mode

### Modifying planned times

The commands used for setting the times in the scheduler window are as follows:

- Click on the day -> Set or delete all the 24 hours of that day of the week
- Click on the hour -> Set or delete that hour in all the days of the week
- Click on the hour and day intersection box - > Set or delete the 24 hours in all the days of the week

One click on an hour box will set the time by a quarter of an hour precisely. An hour is made up of four 15 minute resolutions. Each additional click will change the hour box according to the sequence below:

- one whole hour selection
- the first 15 minutes in the hour selection
- the first 30 minutes in the hour selection
- the first 45 minutes in the hour selection
- the last 15 minutes in the hour selection
- the last 30 minutes in the hour selection
- the last 45 minutes in the hour selection
- deselect the whole hour

## Grid

This type of display allows you to view the planned times of the Scheduler selected using the list-box at the top in the form of a grid. In this case each row in the grid represents an command activation time. A maximum number of rows can be set using the object's 'Max. Nr. Grid Rows' property. Unlike the 'Time Table' you can set times with minute resolutions by entering the exact times you wish in hours and minutes for the activation command start and finish in the two respective 'Start' and 'End' columns directly. Furthermore, you can select the variable, on which the command is to act on, using the 'Tag' column (the command is the one set in selected Scheduler's 'Commands List' and should be a "Variable" command type). You can also set the value in the "Value" column to activate in the variable (ie. if a 'Set' command of a variable, the variable will be set to the value entered in this field).

Scheduler		Add (Ins) Remove (Canc)			
		Start	End	Tag	Value
Monday	1	04:00	04:59		
	2	08:00	08:59		
Tuesday	3	10:00	10:59		
Wednesday	4	21:00	22:59		
Thursday	5				
Friday	6				
	7				
Saturday	8				
Sunday	9				
Save (F2)		Cancel (ESC)		Holidays (F5)	
				Grid Mode (F6)	

A Scheduler Window displayed in 'Grid' mode

### Modifying planned times

To modify the grid just click on the field desired to switch into editing mode if the Scheduler selected allows this.

## Security Options

Scheduler selections using the list-box and any of its modifications can be conditioned by the Access Level settings predefined in the Scheduler. Based on these settings the Scheduler can be set to display in the list-box, and therefore selected, modified or just viewed only. In addition to this, you must consider the variables' "Write Access Level" properties of the Scheduler's set planned times which can be modified. The settings in these properties will determine whether variables can be made available in the "Tag" column for selecting or not.

If the "Always Visible" property is enabled, available for both the Schedulers and Variables ("Access Level" propriety), the Access Level settings will then no longer be considered and

the Scheduler and the Variable in question will always be available for selecting within the "Scheduler Window".

Scheduler objects which are added in runtime are set with a read and write Access Level equal to the access level of the user currently logged in at that moment. In this way the user will be able to edit the scheduler, while other users that do not have equal access levels won't be able to. Schedulers added without any users currently logged in, will be always editable due to the fact that their read and write access levels will be set at "FFFF".



**When the project's Password management is disabled you will need to enable the "Always Visible" property for the Schedulers and Variables you wish to manage in the "Scheduler Window".**

## Hour Selector Style Properties

---

The Style properties from the "Hour Selector" are used for setting the object's graphics properties.

To modify the Style properties, select the object with the mouse and use the Movicon **"Properties Window"**.

### Scheduler Edit Mode

This selection allows you to set the mode with which to display and eventually modify the selected Schedulers time plan. The options are:

- **Time Table:** the scheduled times will be shown in a normal time table where the activation/deactivation ranges can be set graphically with 15 minute resolutions.
- **Grid:** the scheduled times will be shown in a Grid with the option to set activation/deactivation ranges with minute resolutions. In this case you can select the variable to be associated to the Scheduler's command.
- **Both:** in this case the button will be made available in the "scheduler Window" to allow you to switch back and forth from the "Grid" and "TimeTable" modes for maximum use of the time resolutions in order to set more precise time ranges.

Planning in Time Table mode may be less accurate and restricted when selecting 'Both' in the list-box if they work this way in runtime:

From "Time Table" display you can graphically view those areas containing the planned time ranges in "Grid" mode. However, in "Time Table" mode you can only plan those weekdays that do not yet have time ranges set in "Grid" mode. Furthermore, planning in "Time Table" mode may only be done on one day of the week at a time, and only on the day selected in "Grid" mode before switching over to "Time Table" mode. As an alternative you can begin planning in "Time Table" mode then switch over to "Grid" mode to set more accurate time ranges for that day of the week.

### Border

Please refer to **"Style Proprieties common to Drawings and Controls"** for this property's description.

### Save Button

This selection provides the command button for saving any modifications made to the Hour Selector data. Modifications must be made before this action can be executed in Runtime.

### Cancel Button

This selection provides the command button for cancelling any data modifications done to the Hour Selector. This has effect only on data modified after the last save.

### Holidays Button

This selection renders the command button available to allow you switch over from the normal plan to the holiday plan. For further information on "Holidays Planning" please see the appropriate section in Scheduler Chapter.

#### **Scheduler Selection ComboBox**

When enabled, this property, allows you to add a combo-box at the top of the "Scheduler Window" to use for selecting one of the project's schedulers. This list of Schedulers is populated with all those that have been defined in the project and which can be displayed based on their Access Level settings (see "Security Options" for further information), along with any others added in Runtime.

#### **Add Scheduler Button**

When enabled, this property allows you to add a button at the top of the "Scheduler Window" to use for adding new Schedulers while in Runtime mode.

#### **Remove Scheduler Button**

when enabled, this property allows you to add a button at the top of the "Scheduler Window" to use for removing schedulers while in Runtime mode.



**Only Schedulers previously added while in Runtime mode can be removed with the 'Remove' button. Schedulers defined in the project design phase cannot be removed.**

#### **Show Value Column**

When enabled, this property allows a column in Grid mode for selecting the "Set" value to be used in the "Commands ON" of the selected Scheduler. You will then be able to set a different "Set" value for each time frame set in the "Grid". The value set will then replace the one defined in the first "Command on Variable" found in the selected Scheduler's "Commands ON" list. When no value is set, all of the Scheduler's "Commands ON" list will be executed using the settings defined when being planned. If the Scheduler's "Command on Condition ON" list does not contain a "command on variable", there is no need to define a value in the "Grid" columns.

The inserted value will be replaced in the "command on variable's" "Value" field and then used based on the "Action" type defined in the command (Set, Toggle, Increase, ecc.).

#### **Show Variable (Tags) Column**

When enabled, this property allows you to display a column, in Grid mode, for selecting the variable to be used in the "Commands ON" of the selected Scheduler. You can then set a different variable for each time frame set in the "Grid". The variable set will replace the one defined in the first "command on variable" and found in the selected Scheduler's "Commands ON" list. In cases where no variable has been set, all of the scheduler's "Commands ON" list will be executed using the settings defined in its planning stage. If the Scheduler's "Command on Condition ON" list does not contain a "command on variable", there is no need to select a variable in the "Grid" columns.

The list of Variables which can be selected will be populated with all those that have been defined in the project and which can be displayed based on their Access Level settings (see "Security Options" for further information).

#### **Button Size**

Please refer to **"Style Proprieties common to Drawings and Controls"** for this property's description.

#### **Align Buttons**

Please refer to **"Style Proprieties common to Drawings and Controls"** for this property's description on.

#### **Save Button Text**

The text be displayed on the Save button is inserted in this edit box. The default text will be used when left blank.

#### **Cancel Button Text**

The to be displayed on the Cancel button is inserted in this edit box. The default text will be used when left blank.

#### **Holidays Button Text**

The to be displayed on the Holidays button is inserted in this edit box. The default text will be used when left blank.

#### **All Selection Text**

The text to be displayed in the first top left cell in the window is inserted here. The default text will be used when left blank.

**Sunday Text**

The text to be displayed in the '**Sunday**' box is inserted here. The default text will be used when left blank.

**Monday Text**

The text to be displayed in the '**Monday**' box is inserted here. The default text will be used when left blank.

**Tuesday Text**

The text to be displayed in the '**Tuesday**' box is inserted here. The default text will be used when left blank.

**Wednesday Text**

The text to be displayed in the '**Wednesday**' box is inserted here. The default text will be used when left blank.

**Thursday Text**

The text to be displayed in the '**Thursday**' box is inserted here. The default text will be used when left blank.

**Friday Text**

The text to be displayed in the '**Friday**' box is inserted here. The default text will be used when left blank.

**Saturday Text**

The text to be displayed in the '**Saturday**' box is inserted here. The default text will be used when left blank.

**Grid Mode Button Text**

The text you wish to appear on the "**Grid Mode**" button is entered in this editbox. The default text will be used if left blank.

**Add Scheduler Button Text**

The text you wish to appear on the "**Add Scheduler**" button is entered in this editbox. The default text will be used if left blank.

**Remove Scheduler Button Text**

The text you wish to appear on the "**Remove Scheduler**" button is entered in this editbox. The default text will be used if left blank.

**Start Time Column Text**

The text you wish to appear in the first grid column is entered in this editbox, which is the one which will show the command's start time. The default text will be used if left blank.

**End Time Column Text**

The text you wish to appear in the second grid column is entered in this editbox, which is the one which will show the command's end time. The default text will be used if left blank.

**Value Column Text**

The text you wish to appear in the forth grid column is entered in this editbox, which is the one which will show the "Set" value for the variable. The default text will be used if left blank.

**Variable Column Text**

The text you wish to appear in the third grid column is entered in this editbox, which is the one which will show the variable selected for the command. The default text will be used if left blank.

**Max. Nr. Grid Rows**

This property is used for setting the maximum number of rows in the Grid which will correspond to number of time frames that can be set for each day of the week.

**Error String**

The text to appear as string error is entered in this editbox. If the user inserts a wrong interval while editing in "Grid" mode, an error message will show. This error message can therefore be customized using this property. The default text will be used if left blank.

**Selection Color**

This property is used for setting color of the cells in the Hour Selector window for the normal planning.

For further information please refer to "**Color Selection**".

**Holiday Selection Color**

This property is used for setting color of the cells in the Hour Selector window for the holiday planning.

For further information please refer to "**Color Selection**".

## Hour Selector Execution Properties

---

The Execution properties of the "Hour Selector" are used for setting up the Scheduler from which the data to be displayed in the Hour Selector will be used.

To modify the Execution properties, select the object with the mouse and use the Movicon "**Properties Window**".

**Project**

The name of the child project, from which Scheduler objected are to be retrieved and displayed, is entered in this edit box. The current project name will be used when this field is left blank.



Only the name of any existing child project of the current project can be entered in this field.

**Scheduler Linked**

The Scheduler object's name is inserted in this edit box and which will be associated to the Hour Selector for displaying planned schedule data.

**Link to a Network Server**

See paragraph "Drawings and Controls common Execution Properties".

**Network Backup Server**

See paragraph "Drawings and Controls common Execution Properties".

**Commands On Click**

See paragraph on Design and Control "Execution Properties".

**Configuration File**

See paragraph on Design and Control "Execution Properties".

**Generate Unique File Name**

See paragraph on Design and Control "Execution Properties".

**Load**

See paragraph on Design and Control "Execution Properties".

**Save**

See paragraph on Design and Control "Execution Properties".



# IP Camera Viewer

---

The Movicon **"IP Camera Viewer"**, available from the "Advanced Shapes" in the **"ToolBox Window"**, is an object through which images taken by a IP camera can be viewed. This object therefore allows you view a sequence of images in realtime received from a IP camera. This type of camera supports current image downloading through http. Downloading is done using jpg images. Images downloaded from the IP Camera Viewer to Win32 platforms are stored in memory (therefore without using the disk), while temporary files are used when downloading to WinCe platforms (the Temp folder is usually loaded in memory).



The IP Camera Viewer is supported both by WinCE versions and WebClient.



## URL Definitions

---

In order to download images from the IP camera you will need to know which URL to set in the "IP Camera Viewer". Each IP camera comes with its own syntax stated in its accompanying manual to obtain the URL to be used and for configuring it as desired. Normally the IP camera can be configured using an Internet Browser, by simply setting the camera's IP address. Therefore you will need to get the URL to be used in order to only display images in html pages. The URL is the same one which you will have to set in the Movicon IP Camera Viewer.

### Examples of URLs for some video cameras you can find on the market today

#### Axis cameras and video servers:

`http://<servername>/axis-cgi/jpg/image.cgi`

#### StarDot cameras and video servers:

##### StarDot NetCam:

`http://<servername>/netcam.jpg`

**StarDot Express 6 (video server)**

http://<servername>/jpeg.cgi?<camerainumber>

http://<servername>/jpeg.cgi?3

**PiXORD cameras:**

http://<servername>/images<channel><resolution>

http://<servername>/images1sif

**Panasonic cameras:**

http://<servername>/SnapshotJPEG[?Resolution=<resolution>][&Quality=<quality>]

http://<servername>/SnapshotJPEG?Resolution=320x240&Quality=Standard

**D-Link cameras:**

http://<servername>/cgi-bin/video.jpg

**gadget-spot.co.uk cameras:**

http://<host>[:port]/Jpeg/CamImg.jpg

## The IP Camera Viewer General Properties

---

The "IP Camera Viewer" General properties are used for setting the URL to connect to the IP camera.

To modify the General properties select the object with the mouse and use the Movicon **"Properties Window"**.

**IP Camera URL**

The static URL is entered in this property and will be used by the object for getting images from the IP camera. The images are updated according to the refresh time set in the "IP Camera Refresh" property. Usually URLs to be set for most IP cameras found on today's market should follow this syntax sequence type:

"http://<Server Name or IP Address>:[Port Number]/<Image>.jpg"

The URL's single parameters can be found in the IP camera's manual, or by verifying the IP Telecamera's configuration. Names of The name of the image to be inserted (Image.jpg) is usually the default name proposed when saving the image using the Internet Browser.

Please refer to the "URL Definitions" section for further information.

**URL User**

The name of the project user to be used for connecting to the IP Camera. Some IP Cameras may request for authentication in order to download images. In this case the user name and password to be used for authentication are those of the project user set in this field.

## IP Camera Execution Properties

---

The "IP Camera Viewer" Execution properties are used for defining image update parameters.

To modify the Execution properties, select the viewer object with the mouse and use the Movicon **"Properties Window"**.

**IP Camera URL Variable**

You can enter (or select using the "..." browse button to the right) the name of a project string variable. This consents the URL, used for downloading images from the IP camera, to be changed in runtime by simply inserting the URL in the set variable.

**IP Camera Refresh**

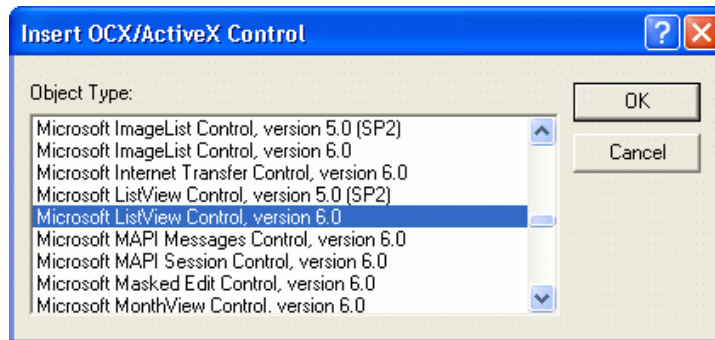
The refresh time, in milliseconds, can be set in this field and which the object will use for updating the IP Telecamera images.

# ActiveX/OCX Objects

The ActiveX/OCX objects are components which can be inserted into any point of the screen and are used for executing operations and functions that are not available in the Movicon system. These objects are actually components of third parties which can be run inside any application that is compatible with the ActiveX/OCX technology.

The ActiveX/OCX are available from the "Special Objects" group found in the **"Objects Window"**.

When selecting the "ActiveX" item the window below will open where a list of the ActiveX/OCX available in the system.



This technique renders the Movicon application open to external object integration where they will be treated as Movicon objects.

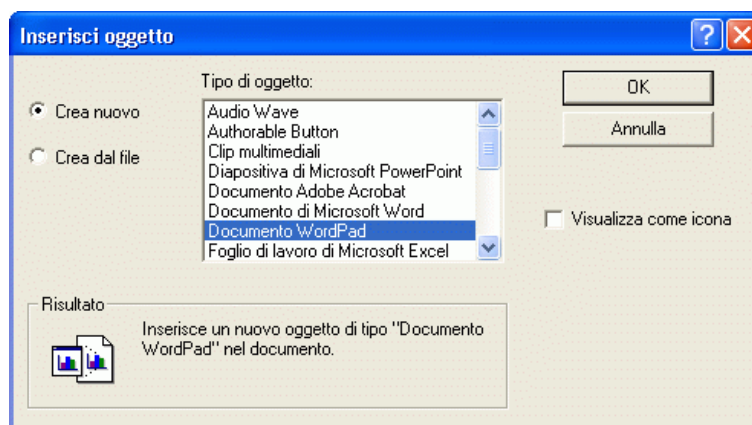
For further information of how to use Active/OCX and their usages please refer to the section titled **"ActiveX Objects"**.

# OLE Objects

The OLE are objects deriving from other applications dynamically linked to Movicon which can be inserted into any point of the screen and are used for executing operations and functions which are not available in the Movicon system.

The OLE objects are available from the "Special Objects" group from the **"Objects Window"**.

When the "OLE" item is selected, the window below will open showing the list of applications available in the system.



This technique renders the Movicon application open for integrating with external applications which will be treated as Movicon objects.

For further information on how to use OLE and its usages, please refer to the section on **"OLE"**.

# Drawings and Controls Common Properties

---

The Movicon Drawings and Controls, which are available in the **"Object Window"**, can be inserted into Screen windows and can be configured and customized through the Movicon **'Properties Window'**.

All Drawings and Controls are vectorial objects and present a variety of different operation analogies, especially concerning Animation properties. Some of their properties are common to all these components.

## Variable Properties

---

The Variables properties of drawing and control objects allow you to associate some of the object's variables for managing its animation and status.

To modify the Variables properties, select the object with the mouse and use the Movicon **"Properties Window"**.

### Default Struct

By using this property you can associate a default structure variable for the symbol.

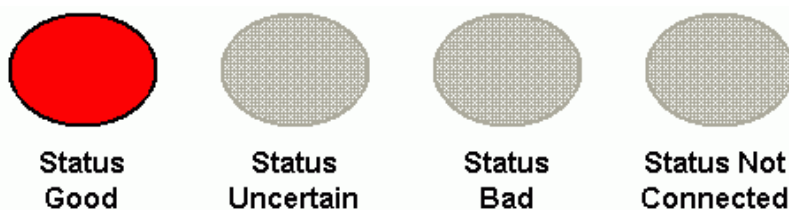
### Enable Var.

This edit box is used for inserting the name of the variable (you can also use the "..."  
browse button on the right hand side to select the variable) to be used for enabling or disabling the component's command functions. When the variable is entered the component will be enabled when the variable's value is different from zero.

The graphic animation functions will also remain active along with IL Logic or script events not User commanded (ie, Events in Variables, SymbolLoading, etc). However, commands that can be evoked by the User will be disabled, being any Command Lists associated to the object of Basic Script code events such as Click, MouseUp, MouseDown, etc.

### Status Var.

This edit box is used for inserting the name of the variable (you can also use the "..."  
browse button on the right hand side to select the variable) whose status (quality) must be displayed graphically by the symbol. The symbol will assume a different graphic state according to the variable's status quality so that an immediate visual understanding of the variable's status can be obtained. For instance, an Elypse inserted with a red background will change as follows according to the variable's status quality:



Only when the quality of the Variable's status is "Good" will the object display normally but differently in all the other cases, generally with a greyish mesh effect colour.



Please keep in mind that the graphical effect may be difference according to the object inserted and the background color it has been setup with.



The graphic management shown above is only active when the "Show Status Mark" property is disabled.

**The same effect obtained by using Win2000/XP systems cannot be obtained with WinCE devices, therefore the status variable graphical animation in symbols is not supported in WinCE.**

However, the "Show Status Mark" property can be used to know whether the status variable quality is Good or not. This function is also supported in WinCE.

## Style Objects Proprieties

---

The Style properties of the drawings or controls elements are used for setting some of their graphical characteristics.

To edit the Style properties, select the object with the mouse and use the Movicon **'Properties Window'**.

Below are described only the style properties common to nearly all of the drawings and controls. Please refer to the appropriate sections for components predisposed with specific properties.

### Clickable (Buttons)

Enabling this check-box will make the selected component clickable. This property is available only in controls such as Buttons, Selection Buttons (Check-box), Option Buttons (Radio Buttons) etc, being those controls already predefined to execute commands when clicked with the mouse.

### Clickable (Windows)

This check box is used to define whether the operator can interact with the Viewer window. When this property is not checked means that the control cannot be managed with the mouse or from the keyboard. In this case, for example, it will be impossible to put columns into order, see any help, execute commands displayed in the window.

This property is only available in Viewer type controls such as the "Alarm Window", "Log Window", "DataLogger Window", "TraceDB Window".

### Auto Layout

When enabling this list layout will be set automatically. This means that the table columns will be resized automatically that all of them can be seen in the Viewer window area.

However when disabling this option the window will open displaying the columns with the sizes set when programmed and therefore the horizontal scroll bar may have to be used for viewing the ones on the right not showing in the window.

This property is only available in Viewer type controls such as the "Alarm Window", "Log Window", "DataLogger Window", "TraceDB Window".

### Show Control Wnd

This property is available only on the objects define as "viewers", such as "Alarm Window", "Log Window", "DataLogger Window", "TraceDB Window" and any "List" object.

Enabling this property you will be able to manage the resizing of the object's columns. Furthermore, in Developing mode, the viewer will become an active window, just like in Runtime mode, so in this case you cannot move it but only resize it.

### Refresh Button

This selection makes the command button operative for refreshing data in the Display window in question. The refresh is carried out according to the active filter's settings.

### Filter Button

This selection makes the command button operative for filtering data to be displayed in the Display window in question. For further information on filter settings, please refer to the following sections:

**"Log Window Data Filter"**  
**"TraceDB Data Filter"**

### Print Button

This selection makes the command button operative for printing data displayed in the Display window in question. The print will be executed by using the report file which must to be specified in the "File Crystal Report" property. Movicon will pass the same filter settings to the Crystal Report for printing the data, which must coincide with that displayed in the Display window in question.

**Report File**

Enter the report file name to be used for printing the data displayed in the Viewer window in question. The Report files, containing the print matrix, can be created with Report Designer or Crystal Reports®. When this field is left blank, Movicon will use the default Report file created by the project.

**Edit Report File**

This command runs the Report Designer for creating or modifying the report to be associated to the viewer window.

**Refresh Button Text**

The text to appear on the "**Refresh Button**" is enter in this edit box. When this field is left blank, the default text will be used instead.

**Filter Button Text**

The text to appear on the "**Filter Button**" is enter in this edit box. When this field is left blank, the default text will be used instead.

**Print Button Text**

The text to appear on the "**Print Button**" is enter in this edit box. When this field is left blank, the default text will be used instead.

**Include Milliseconds**

By enabling this option box milliseconds will also be included in the 'Time' format in the column of the window which supports this type of data.

This property is only available in Viewer type controls such as the "Alarm Window", "Log Window", "DataLogger Window", "TraceDB Window".

**Preserve Aspect Ratio**

This selection allows the proportional dimensions of the object to kept constant. By doing this you will get square geometric shapes instead of rectangle and circular shapes instead of elliptic.

**Button Size**

This setting allows you to select the size of the buttons which are to be displayed in the Alarm Window. The choices are:

- **small**
- **medium**
- **big**

This property is also available for the "Option Buttons" and "Selector Buttons" and an be used for re-proportioning the text size on the control when necessary.

**Align Buttons**

This setting allows you to select the position where the buttons are to be displayed within the Alarm Window. The choices are:

- **left**
- **top**
- **right**
- **bottom**

**Arrow type**

This property is available only for the "Line" and "Connector" object types and allows to define if an arrow should be drawn at the line end. If yes, you can define its placement (left-top, right bottom, both).

**Arrow Height**

This property is available only for the "Line" and "Connector" object types and allows to define the length of the arrow shown if the "Arrow type" property is set.

**Border**

This selection enables the component with a border frame according to the style selected. The border frame will always be in rectangle shape no matter what the shape the object is. The options are:

- none
- bump
- etched
- raised
- sunken

#### Style

This selection allows you to set the control button's display style. A button can in fact be configured with different types of displays, ie. lights or selectors. Keep in mind that this setting is only of a graphic nature.

The Style property is only available for the Button components.

#### Look 3D

This setting allows the component to be displayed in 3D. This property is only available for "Rectangle" objects and makes the Rectangle look like a real button.

#### Look 3D Pressed

This setting allows the component to be displayed in 3D pressed down look. This property is only available for "Rectangle" objects and makes the Rectangle look like a real button.

## Execution Object Properties

---

The drawing and Control execution properties allow you to setup their working characteristics.

To edit the Execution properties, select the object with the mouse and use the Movicon **"Properties Window"**.

Only the Execution properties common to drawings and controls have been described. For those components with special properties please refer to the relevant sections dedicated to them:

#### Network Server

The name or the IP address of the Network Server, where data is to be fetched from and displayed in the window, is entered in this edit box.

#### Network Backup Server

The name or the IP address of the Network Backup Server, where data is to be fetched from and displayed in the window when the primary server is not available, is entered in this edit box.

Connections for Viewers take place after a refresh has been requested and when the Primary Server is not available. When loading pages you need to wait for the connection timeout towards the Main Server to expire before executing a refresh to re-try Backup Server connection.

Each refresh control is commanded as follows:

Data Logger/Recipe Window, DB Trace window, Historical Log Window: "Refresh" button

Alarms Window: "Refresh" method of the "AlarmWndCmdTarget" interface

Graphics: "RecalcLayout" method of the "ChartWndCmdTarget" interface

Trend: Trend's Pause button

Scheduler Window: "Cancel" method of the "HourSelectorCmdTarget" interface



The Backup Server has not been inserted in the child project's properties because, in this case, the Network Server of the variables works as if they were dynamic network variables. The dynamic network variables do not actually manage a Backup Server.

#### Commands on Click

This button opens the Movicon **"Commands List"** through which you can define a list of one or more commands to be executed with a left mouse click on the object (upon releasing the mouse key) or by pressing the "SPACE" or "RETURN" keys when the object is focused on.

For further information about available commands please refer to the paragraph on **"Command List"**.

#### **Ext. File Settings**

In this editbox you can specify the name of the "Configuration Files" to be associated to the object. The object's settings will be saved in the "Configuration Files". You can also use the [...] button to the right of this field to open a window for creating/selecting the file. Both the path and name of the file you wish to associate must be entered. The file is saved with the ".xml" extension. The file name can be defined as pleased or you can use the "Generate Unique File name" command to automatically create a unique name to identity the file for you.



It would be wise to insert the Configuration File into the projects "RESOURCE" folder or in one of its subfolders (or by getting the programmer to setup a new folder if need be).

#### **Generate Unique File name**

When this button is pressed, a unique name for the "Configuration File" will be generated automatically in GUID analog format (number in 128 bit) ie. 0A8DEC92-9213-4DBA-A7E6-C4157ECA8883.xml. This name will only be generated if another name has not yet been entered in the "File di Configurazione" property.



**Attention. This command generates only the file name. You will need to set the path in which the file is to be saved in the "Configuration Files" property.**

#### **Load**

This command loads all the configuration file's settings selected in the "Configuration Files" property and applied to the object.

#### **Save**

This command saves the object's settings in the Configuration File selected in the "Configuration Files" property.

## **General Object Properties**

---

The General properties of drawing and control elements allow you to set a few useful properties for identifying the object.

To edit the General properties, select the object with the mouse and use the Movicon **'Properties Window'**.

#### **Object Name**

This edit box is used for inserting a text string which can be associated as the object's name. Assigning names to objects is useful for various reasons such as the component's description.



It is essential that the object has a name for referential purposes when the Basic Script functions are being used. In this case all the Screen's internal objects must have an unique name.

#### **Object Title**

This edit box is used for inserting a text string to be associated as the object's title. The typed text is also visible in the symbol during Runtime.



When enabling the "Text Properties common to Drawings and Controls" the object's title will be replaced by the variable's contents during Runtime. You can also edit the Title by using the appropriate Basic functions.

#### **Global Name**

This edit box is used for declaring the element's name as a global name on the screen and therefore making it accessible from basic logics from any other drawing by using the "objectname.property" or "objectname.method" syntax.



When this selection is not used, the assigned name will only be acknowledged locally within the symbol.  
The name may be used in any eventual basic script logic in the templates management.

#### **Object Public Name**

This edit box is used for assigning a name which will be managed as a Public Name (for the Screen or the project) through which all the inherent characteristic settings of Drawings/Symbols and Controls will be referred to. These functionalities, described in the appropriate paragraph **"Public Symbols"** permit the automatic editing of all the symbols' properties belonging to a predefined Public Name.



The symbols' inheritance function lets predefined symbol categories be set in order that their properties can be edited in a certain point of the project. For further information on this potentiality please consult the appropriate paragraph on **"Public Symbols"**.

#### **Preserve Size**

When this check-box is enabled the sizes assigned to the component in question will remain unaltered during the Public Symbols updating phase. If this is not selected the component's sizes will be adapted to the parent public symbol, being the one contained in the base Screen which can be selected through the **"Public Source Container"** property from the **"Screen Execution Properties"**.

#### **Preserve Dynamics**

When this check-box is enabled the animation dynamic properties assigned to the component in question, will remain unaltered during the Public Symbols updating phase. If this is not checked the component's animation dynamic properties will adapt to the base public symbol, being the one contained in the base reference Screen, which can be selected through the **"Public Source Container"** property from the **"Screen Execution Properties"**.

#### **Preserve Code**

When this check-box is enabled, any Basic Script code associated to the component in question will remain unaltered during the Public Symbols updating phase. If this is not checked the component's Basic Script code will be updated with the one from the base public symbol, which is the one contained in the base Screen that can be selected by means of the **"Public Source Container"** property from the **"Screen Execution Properties"**.

#### **Preserve Text**

When this check-box is enabled the text (title, name) of the component in question, will remain unaltered during the Public Symbols updating phase. If this box is not checked the text (title, name) of the component will be updated with that of the base public symbol, being the one contained in the base Screen, which can be selected by means of the **"Public Source Container"** property from the **"Screen Execution Properties"**.

#### **Preserve Colors**

When this check-box is enabled the colours of the component in question, will remain unaltered during the Public Symbols updating phase. If this box is not checked the colours of the component will be updated with that of the base public symbol, being the one contained in the base Screen, which can be selected by means of the **"Public Source Container"** property from the **"Screen Execution Properties"**.

#### **Style Source Back Color**

When enabling this property, during the runtime phase, the object will inherit the back color from the Style Source Container for the active styles. For further information about this property please refer to "Style Sources in Symbols".

#### **Style Source Text-Edge Color**

When enabling this property, during the runtime phase, the object will inherit the text-edge color from the Style Source Container for the active styles. For further information about this property please refer to "Style Sources in Symbols".

#### **Style Source Font**

When enabling this property, during the runtime phase, the object will inherit the font for the text from the Style Source Container for the active styles. For further information about this property please refer to "Style Sources in Symbols".

### Show Tooltip

A string text which is to be used as Tooltip during Runtime can be entered in this edit box. The Tooltip, containing the text string, will show when the mouse passes over the component. The tooltip will display for about 5 seconds before disappearing automatically.

### Show Dynamic Tooltip

When this check-box is enabled the tooltip containing information inherent to the that component, will show when passing the mouse over the component, for instance it's name, title, the variables collated to animations and their values etc.



The **"Show Dynamic Tooltip"** function only works when the 'Tooltip' property is not enabled otherwise the tooltip string defined by the programmer will show.

### Rotation Angle

This functionality rotates the object selected. The desired values can be inserted into the edit box or the spin buttons can be used on the right hand side. The rotation is executed on 360° turn and therefore the significant values will be from '0' to 360.

The rotation will be executed clockwise for positive values and anti-clockwise for negative values. The rotation uses the object's baricenter which is selected by means of using the **'Baricenter'** selection box described below.

### Baricenter

This functionality allows you to establish which of the object's baricenter the desired rotation is to be executed. The rotation's baricenter can be selected on the perimeter's fixed positions, considering the perimeter of a hypothetical rectangle which circumscribes to the component when it is a different shape, or it can be set manually by activating the 'Custom' selection which enables the manual settings of the object's X and Y coordinates on which the rotation's baricenter is taken into consideration. These coordinates can be inserted in the **'Baricenter Custom X Pos.'** and the **'Baricenter Custom Y Pos.'** described below. When selecting the baricenter, a little black square will appear in the object indicating its position.

### Baricenter Custom X Pos.

This edit box is used for inserting the X coordinate values of the object's baricenter when the **'Custom'** option has been selected in the **'Baricenter'** property.

### Baricenter Custom Y Pos.

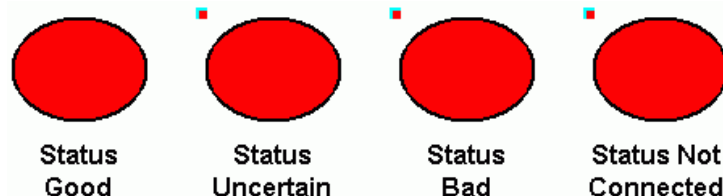
This edit box is used for inserting the Y coordinate values of the object's baricenter when the **'Custom'** option has been selected in the **'Baricenter'** property.

### Open

This button opens the configuration window for controls which have particular functions. For instance the **'Button'** control opens the **'Command List'** window, while a window for selecting the columns to be displayed shows for the "Viewer window" objects.

### Show Status Mark

When this property is enabled, the graphical display of the "Status Variable" associated to a symbol will be managed differently to that described for the "Variable Status" property. A red dot will appear in the top left vertex when its "Status Variable" changes to quality that is different from "good" as follows:



The graphic management shown above is only active when the "Show Status Mark" property is enabled.

### Symbol Description

This edit box is used for inserting the text string which can be used as a description of the selected symbol.

**The 'Description' property is only available for composed Symbols and not for simple drawings and controls.**

#### **Cache Image**

This property consents to enabling or disabling the cache management for composed symbols. When the property is enabled an image of the symbol will be created in the cache, to guarantee the best loading performances. You must, however, keep in mind that this management can only be used for static symbols which do not have any graphical animation. In addition to this, to avoid consuming too much RAM the Windows registry key "**MaxSymbolCache**" can be used to limit the maximum number of symbols whose images can be created in the cache.

**The 'Cache Image' is only available for composed Symbols and not for simple drawings and controls.**

#### **Automatic Enable and Status on Quality**

Movicon consents the automatic enabling or disabling management of objects in screens and their graphical representations using the qualities of the variables associated to them without having to use the "Enable Var." and "Status Var." properties. The **Automatic Enable and Status on Quality Management** allows you to enable/disable this functionality (enabled for default).

The automatic managing of variable qualities associated to controls works in two ways:

##### **Automatic Status Variable Management**

When no variable has been defined in the object's "Status Var." property, Movicon will search through the variables associated to the object and the first one it finds with a valid dynamic part (Fixed I/O address) will be used as the Status Variable to manage the objects coloring based on the variable's quality. When the variable's quality is not set at "Good", the object will change its graphical aspect.

##### **Automatic Enabling Variable Management**

When no variable has been defined in the object's "Enable Var." property, Movicon will search for the first one of the objects associated variables with a valid dynamic part ("Fixed I/O address) and use it as the Enabling Variable. In this case the object's enabling will be managed based on the variable's quality, meaning that it will be enabled when the variable's quality is "Good".

## **Visible Properties Animation**

---

The Visible properties permit you to display the component on the screen in function with the value contained (based on the condition set) in the associated variable.

This property is part of the Drawings and Controls '**Animation**' Properties group.

To edit the Visible properties, select the object with the mouse and use the Movicon '**Properties Window**'.

#### **Enable**

Enabling this check-box will activate the Visible function in the selected component. By doing this the component is displayed or hidden during Runtime in function with the associated Variable's value and the Condition selected.

#### **Variable**

The name of the variable which will determine the components visibility on screen is entered in this edit box (or selected with the "..." browse button on the right hand side).

#### **Value**

The desired value which will then be confronted with the value contained in the Variable to determine the component's visibility is entered in this edit box.

#### **Tolerance**

The tolerance can also be set when a 'equal to' Condition has been selected, being an offset value for the 'equal to' condition. For instance you want to make the component visible when the Value is equal to '5'. If you set the tolerance at '1', in reality the component will become visible when the Variable obtains values 4, 5 and 6.

### Condition

This selection permits you to set the conditions between the Variable and the base Value to determine the component's visibility. The conditions can be:

- <: the component will be visible when the Variable's value is less than the base Value
- >: the component will be visible when the Variable's value is more than the base Value
- =: the component will be visible when the Variable's value is equal to the base Value

## Transparency Animation Property

---

The Transparency property allows the transparency of objects on screen to be managed according to the values contained in their associated variables, allowing their transparency to be modified in runtime as well.

This property belongs to the Drawing and Control **"Animation"** group properties.

To modify the Transparency property, select the object with the mouse and use the Movicon **"Properties Window"**.



**Transparency animation is NOT supported in WinCE.**

**The Transparency property does not have any effect on "Dundas Gauges" and "Chart" object.**

**The "Dundas Gauges" have their own transparencies set in their wizard. The "chart" has its own background management which can be set in gradients and does not support transparency.**

**Furthermore, the transparency is not managed in any objects that rotate ( "General - Rotation Angle" property or "Animation - Rotation" property).**

### Enable Transparency

Enabling this check-box will activate the Transparency management in the selected object. In this way the object will obtain the transparency level according to the associated variable value in Runtime mode.

### Variable Transparency

The name of the variable is entered (or selected using the"..." browse button to the right) in this editbox to determine the object's transparency level. The values include 0 to 255 (0 = completely invisible, 255 completely visible).

## Composed Movement Animation

---

Movicon offers a very handy graphic animation tool to move objects along on not linear trajectories, but composed of two or more straight line segments: **Composed Movement**. The **Composed Movement (XY)** permits a variable to be associated to the movement of the component in the Screen, in function with a path composed of straight line segments which can be freely edited with the mouse.

This kind of editing lets you easily trace out the path which the component will follow in proportion to the associated variable's value by automatically executing interpolations on the X and Y coordinates.

This property is part of the **'Animations'** properties of Drawings and Controls group.

To modify the Composed Movement (XY) properties, select the object with the mouse and use the Movicon **'Properties Window'**.

### Enable

Enabling this check-box will activate the selected component's Composed Movement. By doing this the component will assume the position on the Screen according to the associated Variable's values and the predefined movement trajectory during Runtime.

**Variable**

The name of the variable which determines the component's position along the predefined trajectory is entered in this edit box (or selected by means of using the "..." browse button on the right hand side).

When the variable obtains the maximum value the symbol will go to the end position traced in the trajectory settings. When the variable obtains the minimum value the symbol will go to the start position. The variable's intermediate values will correspond proportionally to the path traced out for the movement.

**Start**

The minimum value that the variable can obtain is entered in this edit box, and will correspond to the traced trajectory's start position. The object will remain at the start position when any Variable values inferior to the minimum Threshold have been obtained.

**End Threshold**

The maximum value that the variable can obtain is entered in this edit box and will correspond to the traced trajectory's end position. The object will remain at the end position when any Variable values superior to the Maximum Threshold have been obtained.

**Edit**

This command allows you to activate and start the component's composed movement editing.

**Reset**

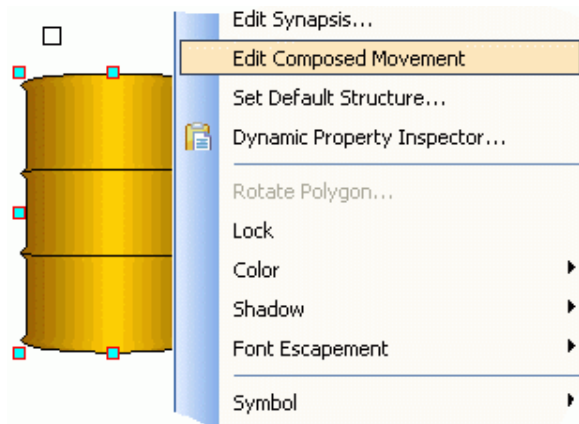
This command resets any composed movements associated to the component. By executing this command any movement trajectories associated to the component will be cancelled.

## **Composed Movement Editing**

---

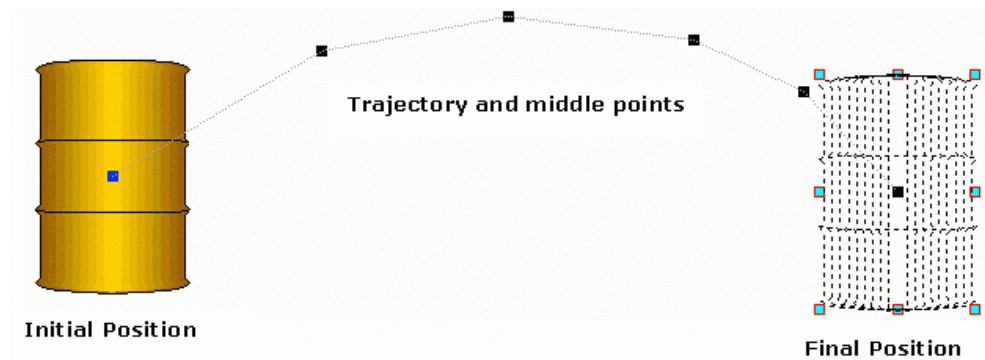
The editing of the Composed Movement involves two phases: the first requires the enabling of the function in the symbol's or control's property described above, specifying the desired variable.

The second phase requires the setting up of the trajectory. To edit the trajectory's settings you need to use the "Edit" command or the **"Edit Composed Movement"** command which can be accessed from the 'Symbol' menu or clicking with the mouse's right button on the component:



After having activated the command a trajectory symbol will appear ready to be positioned on the end point. The procedure is as follows:

1. Position the trajectory symbol on the trajectory's end point
2. Insert the intermediate points by double clicking on the point desired in the Screen. The inserted point will be the second one in respect to the starting position. The following points inserted will follow suit
3. Double click on the third and on all the other additional points which make up the trajectory to be executed. Each point inserted will succeed the previous one
4. Press EXIT to interrupt the composed movement editing



The composed movement permits a composed trajectory to be created for the symbol. The points are inserted one after the other beginning with the starting point and finishing with the ending point. The EXIT key aborts the editing procedure.

Once the 'Composed Movement' has been setup, the set trajectory can be cancelled by simply activating the 'Edit Composed Movement' command again by keeping the 'CTRL' key pressed down.

## Move X Properties Animation

The Move X property permits the selected components to move across the screen along the X axis in function with the values contained in the associated variable (in pixel).

This property is part of the Drawings and Controls '**Animations**' property.

To edit the Move X properties, select the object with the mouse and use the Movicon '**Properties Window**'.

### Enable X Movement

Enabling this check-box will activate the Move X function in the selected component. By doing this the component will assume the positions along the X axis in the Screen according to the associated Variable's values in Runtime.

### Invert X Direction

Enabling this check-box will activate the function to invert the horizontal movement. By doing this, the component will obtain the position according to the below formula in Runtime:

$$Xpos = "End Value X" - "Variable for Move X" Value$$

### Variable for Move X

The name of the variable which will determine the component's position in the X axis is entered in this edit box (or select it by using the "..." browse button on the right hand side).

The value contained in the Variable expresses the component's position on the screen in pixels. When the variable has a minimum value the symbol will be positioned to the point according to the total amount taken from the starting point and the **Start** value. When the variable has a maximum value, the symbol will be positioned to a point according to the quoted sum taken from the starting point and the **End** value.



The position values contained in the base Variable are expressed in pixels, therefore when inserting values higher than the adopted screen resolution, the selected object may exit from the area displayed on the screen.

### Start Value X

The Starting position value, to be obtained by the variable, is entered in this edit box. This value does not represent the absolute position regarding the screen's X axis' '0' coordinates, but the position relating to the coordinates of the component's top left apex concerning the component's start position in the Screen. When the **Variable** values are

lower than the **Start** value the object will be positioned to the point relating to the quota taken from the total amount of the **Start** value and the **Start Offset** value.

#### End Value X

The End position value, to be obtained by the variable, is entered in this edit box. This value does not represent the absolute position regarding the screen's X axis' '0' coordinates, but the position relating to the coordinates of the component's top left apex concerning the component's starting position in the Screen. When the Variable values are higher than the End value the object will be positioned to the point relating to the quota taken from the total amount of the **End** value and the **End Offset** value.

#### Offset for the Start Value X

The position Offset value, which the component is to assume when the **Variable** is set with values lower than the **Start** value, is entered in this edit box. In this case the object will be positioned to the point relating to the quota taken from the total amount of the **Start** value and the **Start Offset** value.

#### Offset for the End Value X

The position Offset value, which the component is to assume when the **Variable** is set with values higher than the End value, is entered in this edit box. In this case the object will be positioned to the point relating to the quota taken from the total amount of the **End** value and the **End Offset** value.

## Move Y Properties Animation

---

The Move Y properties allow you to move the selected component on the screen along the Y axis according to the values contained in the associated variable (in pixels).

This property is part of the Drawings and Controls '**Animation**' property.

To edit the Move Y property, select the object with the mouse and use the Movicon '**Properties Window**'.

#### Enable Y Movement

When enabling this check-box the Move Y function will be activated in the selected component. In this way the component will obtain the position along the Y axis in the Screen in function with the values of the associated Variable in Runtime.

#### Invert Y Direction

enabling this check-box will activate the function to invert the vertical movement. Therefore the component will obtain the position according to the formula below in Runtime:

$$Y_{pos} = \text{"End Value Y"} - \text{"Variable for Move Y" value}$$

#### Variable for Move Y

The name of the variable which will determine the component's position in the Y axis is entered in this edit box (or select it by using the "..." browse button on the right hand side).

The value contained in the Variable expresses the component's position on the screen in pixels. When the variable has a minimum value the symbol will be positioned to the point according to the total amount taken from the starting point and the **Start** value. When the variable has a maximum value, the symbol will be positioned to a point according to the quoted sum taken from the starting point and the **End** value.



The position values contained in the base Variable are expressed in pixels, therefore when inserting values higher than the adopted screen resolution, the selected object may exit from the area displayed on the screen.

#### Start Value Y

The Starting position value, to be obtained by the variable, is entered in this edit box. This value does not represent the absolute position regarding the screen's Y axis' '0' coordinates, but the position relating to the coordinates of the component's top left apex relating to the component's start position in the Screen. When the **Variable** values are

lower than the **Start** value the object will be positioned to the point relating to the quota taken from the total amount of the **Start** value and the **Start Offset** value.

#### End Value Y

The End position value, to be obtained by the variable, is entered in this edit box. This value does not represent the absolute position regarding the screen's Y axis' '0' coordinates, but the position relating to the coordinates of the component's top left apex relating to the component's starting position in the Screen. When the Variable values are higher than the End value the object will be positioned to the point relating to the quota taken from the total amount of the **End** value and the **End Offset** value.

#### Offset for the Start Value Y

The position Offset value, which the component is to assume when the **Variable** is set with values lower than the **Start** value, is entered in this edit box. In this case the object will be positioned to the point relating to the quota taken from the total amount of the **Start** value and the **Start Offset** value.

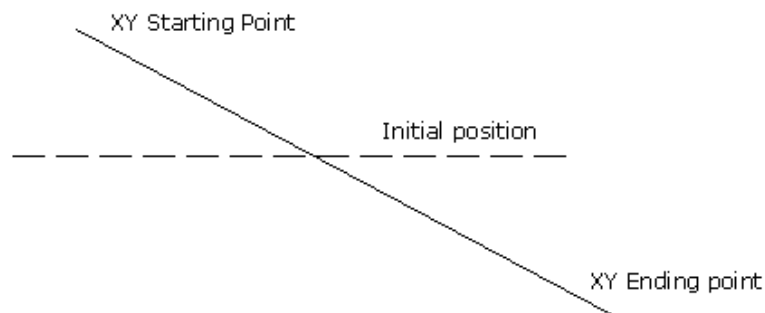
#### Offset for the End Value Y

The position Offset value, which the component is to assume when the **Variable** is set with values higher than the End value, is entered in this edit box. In this case the object will be positioned to the point relating to the quota taken from the total amount of the **End** value and the **End Offset** value.

## Points Properties Animation

---

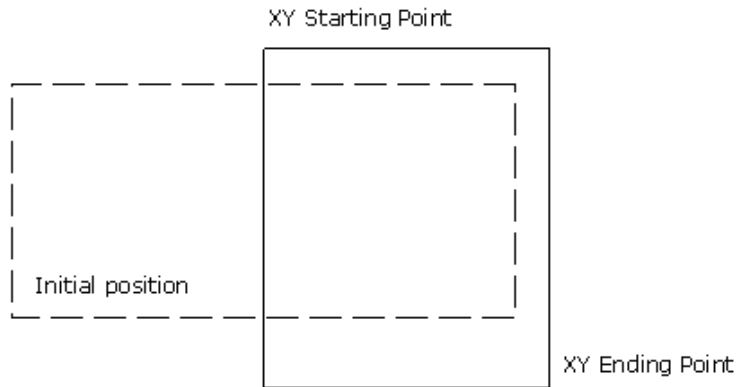
This function allows you to create animations dedicated to dynamically sizing drawings and controls free hand. The objects can receive the XY starting points and XY ending points directly from the Movicon variables. This enables the animation functionalities to be used for determining non linear sizing (therefore different from the '**Scale**' properties) for any kind of necessity. Above all this type of animation can be applied to 'Lines', which become elastically animated.



An example of a line animation: The trajectory line illustrates the line before being moved to the coordinates given by the variable.

When the object is not in line form, the starting point is the top left corner and the ending point is the bottom right corner.





To set the properties concerning points refer to the appropriate sections:

**Start X Point Properties common to Drawings and Controls**

**Start Y Point Properties common to Drawings and Controls**

**End X Point Property common to Drawings and Controls**

**End Y Point Properties common to Drawings and Controls**

## Start X Point Properties Animation

The Start X Point properties allow the component's start X coordinate to move in function with the value contained in the associated variable (in pixels).

The Start X Point is the X coordinate at the start of the line for line type drawings and the coordinate of the far left corner on the screen for flat shapes such as rectangles.

This property is part of the Drawings and Controls '**Animation**' properties group.

To modify the Start X Point property, select the object with the mouse and use the Movicon '**Properties Window**'.

### Enable

When enabling this check-box the Start X Point function will be activated in the selected component. By doing this the component will change the Start X Point in the Screen according to the value of the associated Variable during Runtime.

### Variable

The name of the variable which determines the Start X Point position is entered in this edit box. (or select it by using the "..." browse button).

The value contained in the Variable expresses the Start X Point position in pixels on the screen. When the variable obtains a minimum variable, the symbol's Start X Point is positioned according to the total amount taken from the starting point and the **Start** value. When the variable obtains a maximum value, the symbol's Start X Point will be positioned to a point according to the quoted sum taken from the starting point and the **End** value.



The position values contained in the base Variable are expressed in pixels, therefore when inserting values higher than the adopted screen resolution, the selected object's Start X Point may exit from the area displayed on the screen.

### Start

The Starting position value, to be obtained by the variable, is entered in this edit box. This value does not represent the absolute position regarding the screen's X axis' '0' coordinates, but the position relating to the coordinates of the component's far left apex relating to the component's start position in the Screen. When the **Variable** values are lower than the **Start** value the object's Start X Point will be positioned to the point relating to the difference taken from the total amounts of the **Start** value and the **Start Offset** value.

### End

The End position value, to be obtained by the variable, is entered in this edit box. This value does not represent the absolute position regarding the screen's X axis' '0' coordinates, but the position relating to the coordinates of the far left apex relating to the component's starting position in the Screen. When the **Variable** values are higher than the **End** value the object's Start X Point will be positioned to the point relating to the quota taken from the total amount of the **End** value and the **End Offset** value.

### Start Offset

The position Offset value, which the component's Start X Point is to assume when the **Variable** is set with values lower than the **Start** value, is entered in this edit box. In this case the object's Start X Point will be positioned to the point relating to the quota taken from the difference of the total amounts of the **Start** value and the **Start Offset** value.

### End Offset

The position Offset value, which the component's Start X Point is to assume when the **Variable** is set with values higher than the End value, is entered in this edit box. In this case the object's Start X Point will be positioned to the point relating to the quota taken from the total amount of the **End** value and the **End Offset** value.

## Start Y Point Properties Animation

---

The Start Y Point properties allow the component's Start Y coordinate to move in function with the value contained in the associated variable (in pixels).

The Start Y Point is the Y coordinate at the start of the line for line type drawings and the coordinate of the highest apex on the screen for flat shapes such as rectangles.

This property is part of the Drawings and Controls '**Animation**' properties group.

To modify the Start Y Point property, select the object with the mouse and use the Movicon '**Properties Window**'.

### Enable

When enabling this check-box the Start Y Point function will be activated in the selected component. By doing this the component will change the Start Y Point in the Screen according to the value of the associated Variable during Runtime.

### Variable

The name of the variable which determines the Start Y Point position is entered in this edit box. (or select it by using the "..." browse button).

The value contained in the Variable expresses the Start Y Point position in pixels on the screen. When the variable obtains a minimum value, the symbol's Start Y Point is positioned according to the total amount taken from the starting point and the **Start** value. When the variable obtains a maximum value, the symbol's Start Y Point will be positioned to a point according to the quoted sum taken from the starting point and the **End** value.



The position values contained in the base Variable are expressed in pixels, therefore when inserting values higher than the adopted screen resolution, the selected object's Start Y Point may exit from the area displayed on the screen.

### Start

The Starting position value, to be obtained by the variable, is entered in this edit box. This value does not represent the absolute position regarding the screen's Y axis' '0' coordinates, but the position relating to the coordinates of the component's highest apex relating to the component's start position in the Screen. When the **Variable** values are lower than the **Start** value the object's Start Y Point will be positioned to the point relating to the difference taken from the total amounts of the **Start** value and the **Start Offset** value.

### End

The End position value, to be obtained by the variable, is entered in this edit box. This value does not represent the absolute position regarding the screen's Y axis' '0' coordinates, but the position relating to the coordinates of the highest apex relating to the

component's starting position in the Screen. When the **Variable** values are higher than the **End** value the object's Start Y Point will be positioned to the point relating to the quota taken from the total amount of the **End** value and the **End Offset** value.

#### Start Offset

The position Offset value, which the component's Start Y Point is to assume when the **Variable** is set with values lower than the **Start** value, is entered in this edit box. In this case the object's Start Y Point will be positioned to the point relating to the quota taken from the difference of the total amounts of the **Start** value and the **Start Offset** value.

#### End Offset

The position Offset value, which the component's Start Y Point is to assume when the **Variable** is set with values higher than the End value, is entered in this edit box. In this case the object's Start Y Point will be positioned to the point relating to the quota taken from the total amount of the **End** value and the **End Offset** value.

## End X Point Properties Animation

---

The End X Point properties allow the component's End X coordinate to move in function with the value contained in the associated variable (in pixel).

The End X Point is the X coordinate at the end of the line for line type drawings and the coordinate of the far right corner on the screen for flat shapes such as rectangles.

This property is part of the Drawings and Controls '**Animation**' properties group.

To modify the End X Point property, select the object with the mouse and use the Movicon '**Properties Window**'.

#### Enable

When enabling this check-box the End X Point function will be activated in the selected component. By doing this the component will change the End X Point in the Screen according to the value of the associated Variable during Runtime.

#### Variable

The name of the variable which determines the End X Point position is entered in this edit box. (or select it by using the "..." browse button).

The value contained in the Variable expresses the End X Point position in pixels on the screen. When the variable obtains a minimum value, the symbol's End X Point is positioned according to the total amount taken from the starting point and the **Start** value. When the variable obtains a maximum value, the symbol's End X Point will be positioned to a point according to the quoted sum taken from the starting point and the **End** value.



The position values contained in the base Variable are expressed in pixels, therefore when inserting values higher than the adopted screen resolution, the selected object's End X Point may exit from the area displayed on the screen.

#### Start

The Starting position value, to be obtained by the variable, is entered in this edit box. This value does not represent the absolute position regarding the screen's X axis' '0' coordinates, but the position relating to the coordinates of the component's far right apex relating to the component's start position in the Screen. When the **Variable** values are lower than the **Start** value the object's End X Point will be positioned to the point relating to the difference taken from the total amounts of the **Start** value and the **Start Offset** value.

#### End

The End position value, to be obtained by the variable, is entered in this edit box. This value does not represent the absolute position regarding the screen's X axis' '0' coordinates, but the position relating to the coordinates of the far right apex relating to the component's starting position in the Screen. When the **Variable** values are higher than the **End** value the object's End X Point will be positioned to the point relating to the quota taken from the total amount of the **End** value and the **End Offset** value.

### Start Offset

The position Offset value, which the component's End X Point is to assume when the **Variable** is set with values lower than the **Start** value, is entered in this edit box. In this case the object's End X Point will be positioned to the point relating to the quota taken from the difference of the total amounts of the **Start** value and the **Start Offset** value.

### End Offset

The position Offset value, which the component's End X Point is to assume when the **Variable** is set with values higher than the End value, is entered in this edit box. In this case the object's End X Point will be positioned to the point relating to the quota taken from the total amount of the **End** value and the **End Offset** value.

## End Y Point Properties Animation

---

The End Y Point properties allow the component's End Y coordinate to move in function with the value contained in the associated variable (in pixel).

The End Y Point is the Y coordinate at the end of the line for line type drawings and the coordinate of the lowest corner on the screen for flat shapes such as rectangles.

This property is part of the Drawings and Controls '**Animation**' properties group.

To modify the End Y Point property, select the object with the mouse and use the Movicon '**Properties Window**'.

### Enable

When enabling this check-box the End Y Point function will be activated in the selected component. By doing this the component will change the End Y Point in the Screen according to the value of the associated Variable during Runtime.

### Variable

The name of the variable which determines the End Y Point position is entered in this edit box. (or select it by using the "..." browse button).

The value contained in the Variable expresses the End Y Point position in pixels on the screen. When the variable obtains a minimum value, the symbol's End Y Point is positioned according to the total amount taken from the starting point and the **Start** value. When the variable obtains a maximum value, the symbol's End Y Point will be positioned to a point according to the quoted sum taken from the starting point and the **End** value.



The position values contained in the base Variable are expressed in pixels, therefore when inserting values higher than the adopted screen resolution, the selected object's End Y Point may exit from the area displayed on the screen.

### Start

The Starting position value, to be obtained by the variable, is entered in this edit box. This value does not represent the absolute position regarding the screen's Y axis' '0' coordinates, but the position relating to the coordinates of the component's lowest apex relating to the component's start position in the Screen. When the **Variable** values are lower than the **Start** value the object's End Y Point will be positioned to the point relating to the difference taken from the total amounts of the **Start** value and the **Start Offset** value.

### End

The End position value, to be obtained by the variable, is entered in this edit box. This value does not represent the absolute position regarding the screen's Y axis' '0' coordinates, but the position relating to the coordinates of the lowest apex relating to the component's starting position in the Screen. When the **Variable** values are higher than the **End** value the object's End Y Point will be positioned to the point relating to the quota taken from the total amount of the **End** value and the **End Offset** value.

### Start Offset

The position Offset value, which the component's End Y Point is to assume when the **Variable** is set with values lower than the **Start** value, is entered in this edit box. In this case the object's End Y Point will be positioned to the point relating to the quota taken from the difference of the total amounts of the **Start** value and the **Start Offset** value.

### End Offset

The position Offset value, which the component's End Y Point is to assume when the **Variable** is set with values higher than the End value, is entered in this edit box. In this case the object's End Y Point will be positioned to the point relating to the quota taken from the total amount of the **End** value and the **End Offset** value.

## Scaling Properties Animation

---

The Scaling properties allow the drawing or control to be sized in the screen according to the values contained (in percentages) in the associated animation variable.

This property is part of the Drawings and Controls '**Animation**' properties group.

To edit the Scaling properties, select the object with the mouse and use the Movicon '**Properties Window**'.

### Enable

When enabling this check-box the Scaling function in the selected component will be activated. In this way the component can be resized according to the values of the associated **Variable** during Runtime.



The object can be resized in percentages in respect to the original size set in the programming stage. When the 0% value is reached the object will disappear, as the result of not having any sizes, while the 100% value corresponds to the size with which the object was created in the programming stage. When setting percentages higher than 100% you will get an enlarged object.

### Variable

The name of the variable whose value will be used for changing the component's size is entered in this edit box (or use the "..." browse button on the right hand side to select it). The variable's value will be converted into a percentage value resulted from the scaling between the **Start/End** values and the **Start/End Percentage** value described below.

### Start

The Variable's value corresponding to the object's scaling **Start Percentage** is entered in this edit box. When the Variable obtains a value lower than this value the object will remain sized with the **Start Percentage**.

### End

The Variable's value corresponding to the object's scaling **End Percentage** is entered in this edit box. When the Variable obtains a value higher than this value the object will remain sized at **Start Percentage**.

### Start Percentage

The Scale's Minimum percentage value which the object may reach is entered in this edit box. It will not be possible to make the object smaller than the percentage value set in this property.

### End Percentage

The Scale's Maximum percentage value which the object may reach is entered in this edit box. It will not be possible to make the object bigger than the percentage value set in this property.

### Type

This option box permits you to define the object's size Direction. In this way the drawing can modify its own scale by expanding or shrinking itself in relation to its fixed reference point.

Therefore a symbol indicating the size direction has to be selected.

There can be more than one direction, and the arrows reported in the drop-down list indicate the direction in which the symbol will be resized.



**When choosing the second Direction group, that is the symbols represented with a red circle and titled 'intersection', the object will not be resized but made partly visible based on the set scale factors.**

## Rotation Properties Animation

---

The Rotation property allows the drawing or control to rotate on its own axis in the screen in function with the value contained in the associated animation variable (in rotation angle degrees).

This property is part of the Drawings and Controls '**Animation**' properties group.

To edit the Rotation properties, select the object with the mouse and use the Movicon '**Properties Window**'.

### Enable

When enabling this check-box the Rotation function in the selected component will be activated. By doing this the component can be rotated on its own baricenter in function with the associated **Variable**.



The object's baricenter is set up through the "Baricenter" property in the "**Drawings and Controls common General Properties**" settings. The position of the baricenter can be managed dynamically by enabling the **Enable X point rotation** and **Enable Y point rotation** properties as described below.

### Variable

The name of the variable whose value will be used for executing the component's rotation is entered in this edit box (or selected by using the "..." browse button on the right hand side). The object is rotated on angles of 60 degrees, therefore the variable can assume an interval of values ranging from 0 to 360, being angles of 60 degree turns. Values higher or lower than this range will be executed on a further rotation of the object. For instance the 720 value will take the object to the start position after having completed two turns.

### Start

The minimum value of the **Variable** which will correspond to the object's rotation start is entered in this edit box. When the Variable obtains values lower than this value the object will rotate in the position obtained by subtracting the **Start** and **Start Angle** values.

### End

The maximum value of the **Variable** which will correspond to the end of the object's rotation end is entered in this edit box. When the Variable obtains values higher than this value the object will rotate on the position obtained by the total amount taken from the **End** and **End Angle** values.

### Start Angle

The rotation value which the component must obtain when the **Variable** is set with values lower than the **Start** value is entered in this edit box. In this case the object will be rotated in the position obtained by subtracting the **Start** and **Start Angle** values.

### End Angle

The maximum value which the component must obtain when the **Variable** is set with values higher than the **End** value is entered in this edit box. In this case the object will be rotated in the position obtained by the total amount taken from the **End** and **End Angle** values.

### Enable X point rotation

This check-box has to be enabled in order to dynamically define the X position of the object's baricenter where the rotation is to take place. This position is given by the value contained in the **X point Variable**.

### X point Variable

The name of the variable in which the value identifies the X position of the object's baricenter where the rotation is to take place is entered in this edit box (or selected with the "..." browse button on the right hand side). This value is expressed in pixels and does not represent the absolute point concerning the '0' coordinates of the screen's X axis, but the point relating to the component's furthest left apex, and therefore the components start position in the Screen. If the variable's value is higher than the object's maximum

width the object's will be considered as the X Baricenter the end X coordinate of the object.

#### Enable Y point rotation

This check-box has to be enabled in order to dynamically define the Y position of the object's baricenter to where the rotation is to take place. This position is given by the value contained in the **Y point Variable**.

#### Y point Variable

The name of the variable in which the value identifies the Y position of the object's baricenter where the rotation is to take place is entered in this edit box (or selected with the "..." browse button on the right hand side). This value is expressed in pixels and does not represent the absolute point concerning the '0' coordinates of the screen's Y axis, but the point relating to the component's highest apex, and therefore the components start position in the Screen. If the variable's value is higher than the object's maximum length the object's will be considered as the Y Baricenter the end Y coordinate of the object.

## Text Properties Animation

---

The Text properties allow the associated variable's value to be displayed in the component, replacing the title, whether being text or numeric type.

This property is part of the Drawings and Controls **'Animation'** properties group.

To edit the Text properties, select the object with the mouse and use the Movicon **'Properties Window'**.

#### Enable

Enabling this check-box will activate the Text function in the selected component. In this way the component will display the associated Variable's value replacing its title during Runtime. This functionality is similar to that of the **'Display (Edit box)'** controls existing in the **'Controls'** class in the **"Object Window"**.



Drawing Text properties, and in particular those of Rectangles, can be used for creating 'read only' Display objects.

#### Variable

The name of the variable whose value is to be displayed by the component is entered in this edit box (or selected with the "..." browse button on the right hand side).

#### Format

The format, with which the numeric values presented in the **Variable** are to be displayed, is specified in this edit box. The type of syntax supported is as follows:

`%[flags][width][.precision][{h | l | L}] type`

Each field of the above syntax is a single character or a number indicating a certain option of the selected format. The most simple formats contain only the percent sign and a character indicating the type (eg. %s). The optional fields, which appear before each character indicating the type, control the other aspects of the format.

**flags** Controls the presence of the sign, spaces, decimal characters, hexadecimal or octal or prefixes:

**+**: the sign is put in front of the value (+ o -). Only the - sign is returned for default.

**0**: When the field width is preceded by "0", the "0" characters are used for reaching the number of characters requested

**#**: when used with the "0", "x" or "X" formats, the "0", "0x" or "0X" prefixes are put in front respectively; when used with the "e", "E" or "f" formats it forces the presence of the decimal character no matter what; when used with

	the "g" or "G" formats it forces the presence of the decimal character without cutting off the initial zero.
<b>width</b>	An integer signed value indicates the minimum number of characters to be printed. When the characters to be printed are lower than the number specified, spaces are added until the value provided is reached. When the width value is preceded by "0", the "0" characters are used instead of the spaces.
<b>precision</b>	An integer signed value preceded by the point (.) indicates the number of characters after the decimal character for those formats with decimals or the number of figures to be printed for integer formats.
<b>type</b>	indicates how to interpret the associated variable. The following formats are available:

- d: integer decimal with sign
- i: integer decimal with sign
- o: integer octal without sign
- u: integer decimal without sign
- x: integer hexadecimal without sign. Uses the "abcdef" characters
- X: integer hexadecimal without sign. Uses the "ABCDEF" characters
- e,E: floating point with exponential sign "[-]d.dddd" where d is a decimal figure of the value, dddd are one or more decimal figures of the sign, ddd are exactly three decimal of the exponent and "sign" is + or -
- f: floating point with sign in the form of "[-]dddd.dddd" where dddd are one or more decimals, based on the number's value and on the parameters set in the above mentioned fields.
- g: floating comma with sign in the form automatically selected between the "e" type and "f" type based on the number value and precision requested
- G: as for the "g" type only that it uses the "E" character instead of the "e" character (when necessary)
- s,S: string

In front of the character indicating which type, if this data type allows it, you can put the "h", "l" or "L" prefixes indicating:

- h: short integer (default)
- l,L: long integer

The preset format types are:

- **x**: where the x number identifies the number of figures to be displayed
- **x.x**: where the x number after the decimal point indicates the number of decimal figures to be displayed

When using the 'x.x' decimal figure format its meaning changes according to variable type whether being 'integer' or 'floating point'. When being an 'integer' number, the value can be displayed as a decimal (divided by 10,100,etc.). When being a 'floating point' number, the selected decimal numbers will be displayed.

#### Example 1

If you want to display a value formatted with three figures, select the '**xxx**' Format. The result will be according to how the variable's value is:



Variable Value	Value Displayed
1	001
10	010
100	100

### Example 2

If you want to display a value formatted with two decimal figures, select the '**x.xx**' Format. The result will be according to how the variable's value is:

Variable Integer Value	Value Displayed
1	0.01
10	0.10
100	1.00

Variable Float Value	Value Displayed
1.2345	1.23



When using formats with decimal figures applied to integer Variables, only their display and not their contents can be edited. In practise divisions of 10, 100, etc., is only for graphical displaying purposes and does not influence the real value of the variable in any way.



**When using basic expression in symbols Text Animation Property, it must be set the format "%f".**

## Back Color Properties Animation

The Back Color Properties allow you to vary the selected component's background colour in function with the changes of the associated variable.

This property is part of the Drawings and Controls '**Animation**' properties group.

To Edit the Back Color properties, select the object with the mouse and use the Movicon '**Properties Window**'.

### Enable

Enabling this check-box will activate the Back Color function in the selected component. By doing this the background colour will change in function with the associated Variable's value during Runtime.

### Variable

The name of the variable, whose value is to be used for the background color change in function with the 'Threshold Color' set, is inserted in this edit box.

### Variable Is Alarm Group

The enabling of this property allows you to set the variable used for the animation as the name of the alarm area to be monitored. In this case the name of the Alarm Area is entered in the "Variable" field instead of the variable. In this way the intervention thresholds will result as:

- 0 = No Alarm
- 1 = At least one alarm active
- 2 = At least one alarm ON

This will allow the status of each alarm area to be monitored graphically by setting the 0,1,2 in the back animation properties.

Furthermore, when this option is enabled with a string type variable entered in the "Variable" field, the contents of this string will be used as the name of the Alarm Area to be monitored.

### Edit Back Color List

By using this property you can assign the desired colours for the component's background to the numeric values obtained by the Variable. Click on the "..." button to open the Color

Threshold configuration window. To get further information please refer to the paragraph on "**Colour Threshold Settings in Drawings and Controls**".

#### **Analog Color**

This property enables a gradual change over from one colour to another involving an 'analog' mixture of the two colours where they change over. This gradual change over of color shades can only be carried out when the Variable's two threshold values have been inserted with intermediate values.

## Edge Color - Dynamic Text Animation

---

The Edge-Text Color properties allow you to vary the colours of edges and any texts (title) of the component selected in function with the changes of the variable associated.

This property is part of the Drawings and Controls '**Animation**' properties group.

To edit the Edge-Text Color properties, select the object with the mouse and use the Movicon '**Properties Window**'.

#### **Enable**

When enabling this check-box the Edge-Text Color function will be activated in the component selected. By doing this the component will change the colours of the edges and any displayed texts in function with the values of the associated variable during Runtime.

#### **Variable Color**

The name of the variable, whose value will be used for changing the colours of the edges and any texts in function with the set '**Threshold Colours**', is entered in this edit box.

#### **Variable Is Alarm Group**

The enabling of this property allows you to set the variable used for the animation as the name of the alarm area to be monitored. In this case the name of the Alarm Area is entered in the "Variable" field instead of the variable. In this way the intervention thresholds will result as:

- 0 = No Allarm
- 1 = At least one alarm active
- 2 = At least one alarm ON

This will allow the status of each alarm area to be monitored graphically by setting the 0,1,2 in the back animation properties.

Furthermore, when this option is enabled with a string type variable entered in the "Variable" field, the contents of this string will be used as the name of the Alarm Area to be monitored.

#### **Edit Edge-Text Color List**

This property is used for assigning colours desired for the component's edges and text of the numeric values obtained by the Variable. Click on the "..." button to open the Color Threshold configuration window.

To get further information please refer to the paragraph on "**Colour Threshold Settings in Drawings and Controls**".

#### **Analog Colors**

This property enables a gradual change over from one colour to another involving an 'analog' mixture of the two colours where they change over. This gradual change over of color shades can only be carried out when the Variable's two threshold values have been inserted with intermediate values.

## Filling Properties Animation

---

The Filling properties allow the a coloured filling in proportion to the selected variable's contents in drawings and controls.

This property is part of the Drawings and Controls '**Animation**' properties group.

To edit the Filling properties, select the object with the mouse and use the Movicon '**Properties Window**'.

#### **Enable**

When enabling this check-box the Filling function will be activated in the selected component. By doing this the component will change its filling percentage in function with the **Filling Variable** associated during Runtime.

#### **Variable**

The name of the variable, whose value will be used for changing the component's filling colour in function with the '**Colour Threshold**' set, is entered in this edit box (or selected with the "... " browse button on the right hand side). By doing this the object's filling colour can be changed in function with the value of this variable to have a different colour based on the filling percentage reached (in this case the same variable is to be set both in the **Variable** field and in the **Variable Filling** field).

#### **Edit Filling Colour List...**

By means of this property the numerics obtained by the **Variable**, described above can be assigned with the chosen colours for the component's filling. To open the Colour Threshold's configuration window click on the "... " button. For further information please refer to the paragraph on "**Colour Threshold Settings in Drawings and Controls**".

#### **Analog Color**

When enabling this property you will get a gradual change of one colour to the next obtaining an 'analog' mix of the two colours nearing the changeover from one colour threshold to another. To get the different shades of colour changes you will have to insert intermediate values between the two threshold values in the variable.

#### **Variable Filling**

The name of the variable whose value is need to set the object's filling percentage is inserted in this edit box (or selected by using the "... " browse button on the right).



The minimum and maximum values that the variable can have, described below, are automatically scaled by Movicon to get a filling from 0% to 100% (completely empty, completely full).

#### **Start**

The value of the variable which is to correspond to the 0% filling, meaning no filling, is entered in this edit box.

#### **End**

The value of the variable which is to correspond to the 100% filling, meaning no filling, is entered in this edit box.

#### **Start Offset**

The filling Offset value, which the components must assume when the **Variable Filling** is set with values lower than the one entered in the **Start** property, is entered in this edit box. In this case the object will be filled with a percentage in proportion to the difference between the **Start** value and **Start Offset** value.

#### **End Offset**

The filling Offset value, which the components must assume when the **Variable Filling** is set with values higher than the one entered in the **End** property, is entered in this edit box. In this case the object will be filled with a percentage in proportion to the difference between the **End** value and **End Offset** value.

#### **Type**

This option box is used for setting which direction the filling is to take when applied. The choices are:

- none
- left-right
- right-left
- bottom-top
- top-bottom

The **"none"** option means that no filling will take place.

#### Start Variable

The name of the variable to be used as a dynamic Start value is inserted in this edit box (or selected with the "..." browse button on the far right). When a variable is entered here, the **Start** property will be ignored and the value contained in the selected variable will take its place.

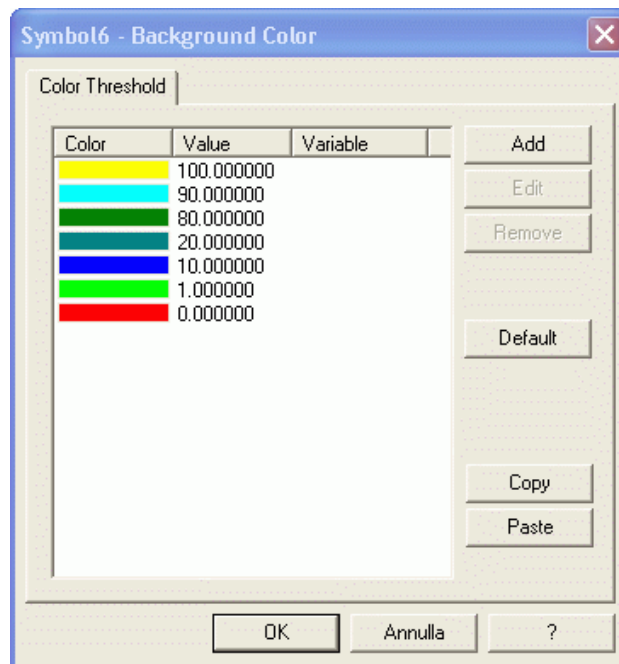
#### End Variable

The name of the variable to be used as a dynamic End value is inserted in this edit box (or selected with the "..." browse button on the far right). When a variable is entered here, the **End** property will be ignored and the value contained in the selected variable will take its place.

## Colour Threshold Settings

The Threshold Settings in Drawings and Controls must be configured when the **"Back Color Properties common to Drawings and Controls"** or **"Edge - Text Color Properties common to Drawings and Controls"** or **"Filling Properties common to Drawings and Controls"** properties are enabled. In this case it is necessary to execute the association of colours to values that the variables, defined for animation, can obtain. The colour thresholds that can be inserted are virtually unlimited.

The associated colour will be activated when the value of the variable obtains the same value defined for the threshold. When the value of the variable exceeds the value defined for the threshold, the colour associated to the next threshold will be activated and so on.



For instance, taking into consideration the settings reported in the above figure we should get:

Variable Value	Colour
0	Red
1	Green
10	Blue
...	...
90	Cyan
100	Yellow
...	...



If the '**Analog Colours**' option has been selected the pass over from one colour threshold to another will be gradual, and therefore a series of intermediate colours managed by Movicon will be displayed during Runtime.

By using the "**Copy**" and "**Paste**" commands described below you can copy the defined color threshold list from one object and paste it in other different objects. This allows you transfer color thresholds from one object to another quickly for all types of dynamic animations: "**Background color**", "**dynamic Text**" and "**Text & Edge Colors**" and "**Gradual Filling**".

The buttons positioned on the right hand side of the dialog window allow you to edit the colour thresholds. Their functions are:

- **Add**: adds a new Colour Threshold. A 'Edit Colour Threshold' window opens
- **Edit**: after having selected a threshold from the list it's settings can be edited. A 'Edit Colour Threshold' window opens
- **Remove**: after having selected a threshold from the list it can be deleted
- **Default**: when pressing this button the threshold list will be cancelled and the 7 default thresholds will be inserted by Movicon
- **Copy**: consents you to copy the color threshold list to the Windows Clipboard. The color threshold data is copied to the Window clipboard in xml format
- **Paste**: consents you to paste the color threshold list, previously copied to the Windows Clipboard with the "copy" command, to the selected object



It is compulsory to define at least two thresholds, one for colouring and the other for restoring the colour (or text).

## Edit Threshold Colour

---

When you press the Add or Edit buttons in the window listing the inserted thresholds, a dialog window will open to configure the parameters of each threshold:

### Value

In this edit box the value desired is entered which then be confronted with the value contained in the animation **variable** to determine when the threshold in question must be activated.

### Variable

In this edit box the name of the variable to be used in place of the **Value** property is to be entered (or select with the "... " browse button on the right). By doing this the threshold's value can be made dynamic, as the contents of the variable will be used and which can be edited during Runtime.



Be careful not to confuse the **Threshold Variable** with the Animation **Variable** inserted for Colour Threshold management. The **Threshold Variable** is only needed to make the value of each single threshold dynamic. This means that if the same variable is inserted, in the **Threshold Variable** field and in the **Variable** field by mistake, the animation will stop working and the components will keep the same colour as the reference variable and the threshold value will remain the same.

### Mode

The mode option allows you to select the animation's behaviour when the threshold has been reached. The following actions, which will be executed when the value has been reached, can be selected:

- **Normal**: when the threshold has been reached the colour, selected by means of the **Color** property described below, will be displayed
- **Invisible**: when the threshold has been reached the colour will become invisible, as if transparent
- **Slow Blinking**: when the threshold has been reached the slow blinking will be executed with the colour defined in the **Color** and **Blinking Color** properties as described below. The blinking time is 2000 msec. for default. This value can be customized using the appropriate "BlinkTypeSlow" registry key.

- **Average Blinking:** when the threshold has been reached the normal blinking will be executed with the colour defined in the **Color** and **Blinking Color** properties as described below. The blinking time is 1000 msec for default. This value can be customized using the appropriate "BlinkTypeMedium" registry key.
- **Fast Blinking:** when the threshold has been reached the fast blinking will be executed with the colour defined in the **Color** and **Blinking Color** properties as described below. The blinking time is 500 msec. for default. This value can be customized using the appropriate "BlinkTypeFast" registry key.
- **Custom Blinking:** when the threshold has been reached the blinker will be executed with the colour defined in the **Color** and **Blinking Color** properties as described below, using the value specified in the **Blink Time**, described below, as blinking frequency

#### **Blink Time**

The blink time is entered in this box which will be used when the **Customized Blinking** item has been selected in the **Mode** property. The time is expressed in milliseconds.

#### **Text**

The text string is entered in this edit box. The string will be displayed inside the component together with any animations that have been set when the preset threshold has been reached. This option allows you to create dynamic texts. This property is only available for the "**Edge - Text Color Properties common to Drawings and Controls**" function.

#### **Color**

This setting is used for selecting the colour to be associated to the Threshold. For further information on selecting colours please refer to the paragraph on "Color Selection".

#### **Blink Colour**

This permits you to select the blink colour to be used together with the threshold's **Color** when the **Customized Blink** in the **Mode** property has been selected. For further information on selecting colours please refer to the paragraph on "Color Selection".

## **Images Properties Animation**

---

The image properties permit one or more images to be displayed in the component based on the value of the variable connected.

This property is part of the Drawings and Controls '**Animations**' property group.

To edit the Image properties, select the object with the mouse and use the Movicon '**Properties Window**'.

#### **Enable**

When enabling this check-box the Image function on the selected component will be activated. This will display the image associated to the value of the selected Variable during Runtime.

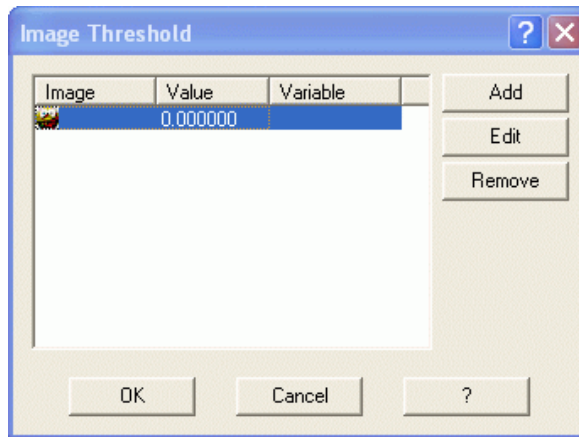
#### **Variable**

The name of the variable, whose value must be associated to the image to be displayed in the component, is entered in this edit box. (or selected with the '...' browse button on the right

#### **Edit Image List**

By using this command you can edit the image list which is to be displayed based on the value of the **Variable**.

The first window to be opened for editing the images shows the list of images already inserted and the corresponding values of the Variable:



When the Variable assumes the value associated to the image, it will be displayed in the component.

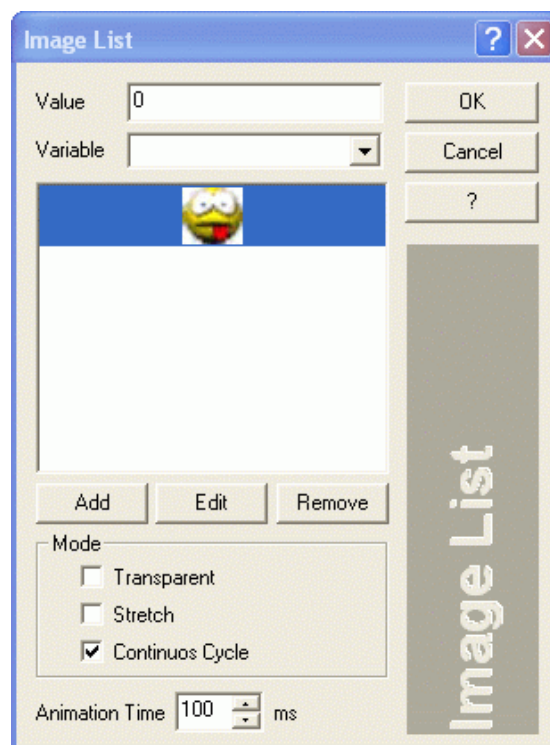
The buttons available for editing the list are:

- **Add:** consents the entry of a new animation on the list (image associated to the Variable's value)
- **Edit:** consents the editing of an animation already inserted on the list. The image to be modified must first be selected.
- **Remove:** consents you to delete the selected image from the list.



At least two thresholds must be set to animate images.

When pressing the '**Add**' and '**Edit**' keys another window will open where the images can be selected and associated with the value of the variable:



This window, called the Image list, permits you to set the image display conditions. This window is used for setting the **Threshold value** referring to the associated **Variable**. The threshold value determines the activation and appearance of the image (or sequence of images), and can be set in fixed constant, 'Value' field, or dynamic value, 'Variable' field.

The buttons used for managing the images have the following functions:

- **Add:** used for selecting a new image
- **Edit:** used for editing images already inserted on the list. The image must first be selected then edited.
- **Remove:** deletes the selected image from the list.

You can associate more than one image to each threshold value. In this case, when the threshold is activated the images are displayed in sequence, in time intervals specified in the '**Animation Time**' field.



The possibility to enter more than one image for each threshold consents the different images to appear in sequence within the symbol by simply creating graphic animation in single sequences or continuous cycle.

When adding the preset threshold, the image of sequence of images will be displayed according to the set mode selected from the following options:

- **Transparent:** the image's colour selected through the "**Transparent**" property from the Fill Attributes Properties common to Drawings and Controls' group will appear transparent in the object
- **Stretch:** the image will be adapted to the preset sizes of the object containing it.
- **Continuous Cycle:** the images will be cycled by overlapping each other continuously until no longer permitted by the activation conditions. Otherwise, the sequence of images will be carried out once only upon the rising edge of the threshold's condition.
- **Animation Time:** if more images have been added in this window, they will be made to automatically appear by Movicon in sequence of entry order, with time intervals (in milliseconds) set in this edit box.

## Visibility Properties

---

The Visibility properties consent the component to be visible on screen in function with the layer activated in the Screen.

To edit the Visibility properties, select the object with the mouse and use the Movicon '**Properties Window**'.

### Layer

When enabling this check-box the Visibility function of the selected component will be activated. By doing this the component will be visible or hidden during Runtime in function with the layer activated in the Screen.

### Enable Zoom

When enabling this property the visibility of the object will automatically be managed when a zoom takes place. When zooming out to reduce the Screen's sizes, the object when reaching a size lower than the percentage expressed in the '**Zoom-size Percent level**' property, will become completely invisible.

### Enable Size

When enabling this property, the object's visibility will automatically be managed when it is being resized. When reaching a size lower than the percentage expressed in the '**Zoom-size Percent level**', while being reduced, it will become completely invisible.

### Zoom-size Percent level

The percentage taken from the object's original sizes. When the object's size goes below this percentage, it will be made invisible when the conditions enabled by the '**Enable Zoom**' and '**Enable Size**' activate.

### Visible on Web Client

The component will also be made visible on the WebClient when this check-box is enabled with a check mark. When this property is disabled not only will the control not be visible



on the WebClient but its basic script code will not be initialized when loaded in a WebClient session as well.

#### **Visible on CE Platform**

When this check-box is checked the component will also be made visible on WinCe platform. When this property is disabled the control, apart from not being visible in WinCE, will not initialize its basic script code when the container screen is loaded.



This property will only be visible when the project is NOT set as "WinCE" in the project's "Platform" property.

#### **Focus Hilite**

When this check-box is enabled the object will invert its background and foreground colors when focused on.

#### **TAB Selectable**

When this check-box is enabled, the object can be selected by using the TAB key during Runtime mode.

#### **Invert Color**

This property inverts the object's back colors and border colours.

#### **Show Focus**

This property is used for enabling the display of the control's outline when focused on.

#### **Show Hilite**

This property is used for enabling the control's inside border to highlight when the mouse pointer is on top of it.

## **Position Properties**

---

The Position properties allow you to read/write the component's position and size on screen.

To edit the Position properties, select the object with the mouse and use the Movicon **'Properties Window'**.

#### **X**

This property expresses the X coordinate of the component's furthest left corner edge. The value is expressed in pixels and relates to the Screen window's 0 point (the top left corner edge) which contains the component. The value of this property will be automatically modified each time the component is moved in the Screen and vice-versa by modifying this value the component will graphically change its position.

#### **Y**

This property expresses the Y coordinate of the component's highest corner edge. The value is expressed in pixels and relates to the Screen window's 0 point (the top left corner edge) which contains the component. The value of this property will be automatically modified each time the component is moved in the Screen and vice-versa by modifying this value the component will graphically change its position.

#### **Width**

This property expressed the component's width. The value is expressed in pixels. The value of this property is automatically modified each time the component is resized in width and vice-versa by modifying this value the component will graphically change its size.

#### **Height**

This property expresses the component's height. The value is expressed in pixels. The value of this property is modified each time the component is resized in height and vice-versa by modifying this value the component will graphically change its size.

## Stroke Attributes Properties

---

The Stroke Attributes Properties allow you to set the colour and sizes of the components' edges and texts.

To modify the Strokes Attributes property, select the object with the mouse and use the Movicon 'Properties Window'.

### Color

This property allows you to select the colour to be associated to the edges and any text displayed by the component.

For further information on selecting colors, please refer to the paragraph on "Color Selection".

### Pen Size

This property allows you to select the size of the component's border lines.

### Pen Style

This property lets you select the component's borderline style.

## Background Attributes Properties

---

The Fill Attributes properties allow you to set the background colours of components.

To edit the Fill Attributes Properties, select the object with the mouse and use the Movicon '**Properties Window**'.

### Brush Style

This property allows you to select the brush style to be applied to the component's background. The list shows a variety of brush style designs available.

### Back Color

This property allows you to select the background color to be applied to the component.

For further information on selecting colours please refer to the paragraph on "Color Selection".

### Gradient Type

This property allows you to select the type of gradient for the background colour to be applied to the component. You can keep the colour uniform by selecting the '**none**' option, and you can also create colour graduations by selecting one of the other options available. The Gradient, when activated, will be a mixture of two colours chosen in the '**Back**' and '**Gradient Color**' properties.

### Gradient Color

This property allows you to select the second colour with Movicon will use to create the components background gradient.

For further information on selecting colours please refer to the paragraph on "Color Selection".

### Filling Color

This property allows you to select the filling's colour to be used when the "**Filling Properties common to Drawings and Controls**" has been used. In this case the colour selected for executing the component's filling will be used only when a "**Variable**" has not been selected in the "**Filling Properties common to Drawings and Controls**".

For further information on selecting colours please refer to the paragraph on "Color Selection".

### Static Image

By using this box you can associate a static BMP, JPG, GIF, WMF, etc. design type as the object's background. As alternative to typing in the path and name, you can select the file using the browse ("...") button to the right of the editbox.



Pressing the "CTRL + (...)" keys will allow you edit the selected background image, when opening the preset application associated to that image in a new window.



To avoid getting any problems with the images' absolute paths, it is advised to insert all the images used in the project inside the 'IMAGES' folder within the project folder.

### Stretch Image

This property is used for setting the sizes of the associated image to fit in the object containing it. Keep in mind, however, that when changing the image's sizes it's graphical look may be effected.

### Image Transparent Color

This property allows you to select a colour which will become transparent in the image associated to the control. The '-1' value consents the colour transparency to be used for those image formats supporting this property.

For further information on selecting colours please refer to the paragraph on "Color Selection".

### Transparency Level

This property allows you to set the selected object's transparency level. The values which can be used start from 0 to 255 (0 = completely invisible, 255 = completely visible).



The settings of this property will be ignored if the "Transparency" Animation property has been enabled which consents to managing the transparency dynamically based on variable values.



**Transparency animation is NOT supported in WinCE.**  
**In addition to this, the transparency property has no effect in "Dundas Meter" and "Chart" objects. The "Dundas Gauges" transparency is set in the object's wizard. The "Chart" has a background attribute management where can be set with a gradient type and color but does not support transparency.**

## Access Levels Properties

---

The Access Levels properties allow you to associate the components with access and user levels so that only the users with the necessary rights can use them.

To edit the Access Levels Properties, select the object with the mouse and use the Movicon **'Property Window'**.



The Access Levels properties are active only when the "Enable Password Manager" property from the **'Users and User Groups General Properties'** has been enabled in the project.

### Write Access Level

By means of this property you can define the Access Level mask needed to execute, for example, the command list associated to the control. If the access level mask of the user logged on at that moment does not correspond to the control's settings, the user will not be able to execute any command operations associated to that control.

For further information see the paragraph on **"User Levels and Access Levels"**.

### Read Access Level

By means of this property you can define the Access Level mask needed for reading the control. If the access level mask of the user logged on at that moment does not correspond to the control's settings, the user will not be able to see the control which becomes invisible.

For further information see the paragraph on **"User Levels and Access Levels"**.

### User Level

By means of this property you can define the User Password level needed for example to execute the commands associated to the control. When the control in question is used by a user, Movicon will request activation of a user with a User Password level the same or higher than the one defined in the control itself. If the user who executed the Log on has

the necessary access rights they will be authentically acknowledged, otherwise the Log on will fail and it will not be possible to carry out the operations requested by the user. For further information see the paragraph on "**User Levels and Access Levels**".

## Font Properties

---

The Text properties allow you to display the associated variable's value, numeric or text, replacing the title of the object.

This property is part of the Drawings and Controls '**Animation**' properties group.

To edit the text properties, select the object with the mouse and use the Movicon '**Properties Window**'.

### List Font

By pressing the '...' button on the right hand side of the box a window will be activated for selecting the font to be associated to the texts presented in the 'Viewer' window. The selection is carried out according to the Windows modality standards.



This property is only available for some components, in particular for the display windows such as the Alarm Window, Log Window, etc.

### Scale Font

By pressing the '...' button on the right of the box a window will be activated for selecting the font to be associated to the texts which represent the scale display of components such as Gauges, Trends, Data Analysis. The select is carried out according to the Windows modality standards.



This property is only available for some components, in particular those representing volume amounts on scales, such as Gauges, Trends, Data Analysis.

### Grid Font

By pressing the '...' button on the right of the box a window will be activated for selecting the font to be associated to the texts which represent the columns of the "Grid" object. The selection is carried out according to the Windows modality standards.

### Adapt Font

When this check-box is enabled the text in the selected component will be activated. By doing this the component will display the values of the associated variable, where the title is, during Runtime. This functionality is very similar to the controls' '**Display (Edit box)**' residing in the '**Controls**' class of the "**Object Window**".



The drawing's Text property, and in particular the Rectangle's, can be used for creating Display objects for reading only.

### 3D Font

This box is used for selecting the type of three-dimensional effect to be associated to eventual texts entered in the title by using the 'General' properties.

The options are:

- **None**
- **Raised**
- **Embossed**

The default value is "None".

### Text Align

This box is used for selecting the type of alignment for the eventual text entered in the title using the 'General' properties.

The options are:

- **Center**
- **Top**
- **Bottom**
- **Left**
- **Right**

#### **Title Font**

The name of variable whose value is to be displayed by the component is entered in this edit box (or select name by using the "..." browse button on the right hand side).  
The default settings are:

Char type: Tahoma  
Style: Normal  
Points: 8

## Script Debug Output Properties

---

The Script Debug Output properties allow the texts inserted in the Basic Script's 'Debug.Print' instructions to be written on log files or printed.



This group of properties is only available if the Drawings and Controls have been associated with the Basic Script codes.

To edit the Script Debug Output properties, select the object with the mouse and use the Movicon **'Properties Window'**.

#### **Status Bar**

When this property is enabled, the indications from the Basic Script 'Debug.Print' function will be shown on the Movicon status bar and in the project's 'Output Window'.

#### **App.Log**

When enabling this property, the indications from the Basic Script 'Debug.Print' function will be reported in the projects historical log.

#### **Spooler**

When enabling this property, the Basic Script 'Debug.Print' function will be sent to the system's print spooler.

## Dragging Properties

---

The Dragging properties allow you to move or resize symbols using the mouse during the runtime phase. The relative positioning and sizing values will be placed in the associated variables.

This property is part of the "Animation" properties for drawings and controls.

To modify the Dragging properties, select the object with the mouse and use the Movicon **"Properties Window"**.



**Object Dragging is not supported by WinCE or Web Client and when objects are in Embedded Screens.**

#### **Move**

The enabling of this property allows object and symbol to be dragged within the screen area during runtime mode. The position will determine the dynamic adaptation of the drawings coordinates in the associated variables.

The coordinates are always in X0 and Y0 pixels coinciding with the top left border of the object, or the rectangle which circumscribes it, in respect the point on the top left of the screen.

#### **Resize**

The enabling of this property allows the object or symbol to be resized by having its borders dragged within the screen area during runtime mode. The resizing will determine the dynamic adaptation of the object's size coordinates in the associated variables.

**X Variable**

The name of the variable, where the value is to be kept updated by Movicon according to the X coordinate (expressed in pixels) of the symbol's position, is entered in this edit box (or selected by using the "..." browse button.

**CX Variable**

The name of the variable, where the value is to be kept updated by Movicon according to the symbol's width value (expressed in pixels), is entered in this edit box (or selected by using the "..." browse button.

**Y Variable**

The name of the variable, where the value is to be kept updated by Movicon according to the Y coordinate (expressed in pixels) of the symbol's position, is entered in this edit box (or selected by using the "..." browse button.

**CY Variable**

The name of the variable, where the value is to be kept updated by Movicon according to the symbol's height value (expressed in pixels), is entered in this edit box (or selected by using the "..." browse button.

**X Min.**

This edit box is used for entering the X coordinate's minimum value (in pixels) within which the operator can move the object or symbol.

**X Max.**

This edit box is used for entering the X coordinate's maximum value (in pixels) within which the operator can move the object or symbol.

**CX Min.**

This edit box is used for entering the minimum value of the X size (in pixels) within which the operator can resize the object or symbol.

**CX Max.**

This edit box is used for entering the maximum value of the X size (in pixels) within which the operator can resize the object or symbol.

**Y Min.**

This edit box is used for entering the Y coordinate's minimum value (in pixels) within which the operator can move the object or symbol.

**Y Max.**

This edit box is used for entering the Y coordinate's maximum value (in pixels) within which the operator can move the object or symbol.

**CY Min.**

This edit box is used for entering the minimum value of the Y size (in pixels) within which the operator can resize the object or symbol.

**CY Max.**

This edit box is used for entering the maximum value of the Y size (in pixels) within which the operator can resize the object or symbol.

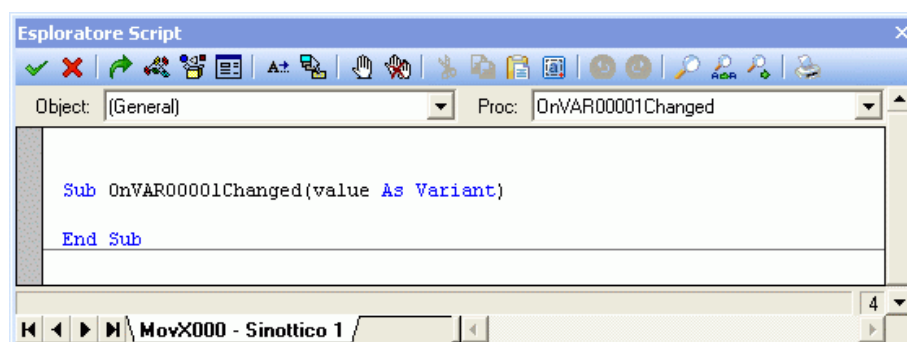
# Associating Variable Script Events to Symbols

---

You can insert customized events, associated to the variations of variables from the Movicon Real Time DB, in addition to the standard ones made available to you by Movicon (Click, DbClick, KeyDown, etc.) internal Symbols' Script Codes (Drawings or Controls). An event can be added to be executed every time the selected variable changes state. Therefore the programmer can decide which code to insert internal the event as deemed necessary. The inserted event will then be active and processed when the Symbol is active, due to it being loaded in Ram.

The association of a Variable Script event to a Symbol can be done by selecting the Symbol and activating the "Add New Variable Script Event" command which can also be found in the Project Explorer's 'Commands' window.

The request for associating a Variable Script Event to a Symbol will open up another window containing the list of available variables from the Real Time DB. Once the variable has been selected, the new event with the "OnNomeVariabileChanged" syntax will automatically be inserted internal the symbol's script code:



# Public Symbols

---

Movicon has a extremely interesting potentiality in the symbol management: the public symbols.

By using the public symbols management you can setup a direct link between different symbol properties, used in different Screens throughout the entire project. The management of this linkage gives you the possibility to automatically set, for all the symbols linked up to each other, the same property or code of the reference symbol.



A typical example of using this type of function would be to create different Screens which all use the same symbol. Let's take a plant valve used in all the project's Screens as an example where, after project completion, the valve's symbol must be edited. By modifying the symbol you are left with the possibility to apply the same changes to all the project's symbols with just one operation.

Movicon permits you to manage any editing needed for updating all the project's drawings or symbols declared as Public Symbols.



In order that a drawing or symbol to become 'public' and subject to global changes, it needs to receive a name in the "Public Name" property of the **'General Properties common to Drawings and Controls'** of the symbol/drawing itself.

It is advised to assign the **"Public Name"** to the symbol's or drawing's property when editing the template (or initial symbol). The name will always be kept associated to the symbol's property, whether it is copied and pastes, saved in the library or taken out later. All the copied symbols will therefore have the same Public Name, making any eventual global editing easier.

## Public Symbols Global Editing

---

The following condition must be preset in order to execute the updating functionalities of Public Symbols:

1. All the public symbols of the same group must have the same **"Public Name"**
2. A reference symbol must be set up from which all the others will get their properties
3. Each symbol has to be established with properties which will remain unaltered and properties which will be updated in accordance with the reference symbol



The updating of public symbols is only carried out in the Runtime phase, therefore during the programming phase, the symbols defined as public will display their native properties and characteristics. Movicon will only carry out any changes made to them during Runtime. This allows the public symbols to be deactivated and to return back to their original configuration.

The Symbols to be subjected to any editing through the Public Symbols must be assigned a **"Public Name"** which is to be the same for all the symbols belonging to the same group. At this point, in order to establish the reference symbol it is necessary to set which is the Screen Containing the reference symbol in the **"Public Source Container"** property of the **'Screen Execution Properties'** group. By doing this the public symbols will be updated with the characteristics of the symbol with the same public name presented in the specified "Public Source Container".

The properties to be kept unaltered and those to be updated in accordance to the reference symbol can be set for each symbol singularly through the **"Drawings and Controls common General Properties"**. The properties involved are as follows:

- **Preserve Size:** the sizes assigned to the component in question will be kept unaltered during the updating phase of the Public Symbols. If the check-box is not selected with a tick, the component's sizes will be adapted to the those of the reference public symbol



- **Preserve Dynamics:** the dynamic animation properties assigned to the component in question will be kept unaltered during the updating phase of the Public Symbols. If the check-box is not selected with a tick, the components dynamic animation properties will be adapted to those of the reference public symbol
- **Preserve Code:** the Basic Script code associated to the component in question will be kept unaltered during the updating phase of the Public Symbols. If the check-box is not selected with a tick, the component's Basic Script code will be updated with that of reference public symbol's
- **Preserve Text:** the text (caption, name) of the component in question will be kept unaltered during the updating phase of the Public Symbols. If the check-box is not selected with a tick, it will be updated with that of the reference public symbol's
- **Preserve Colors:** the colours of the component in question will be kept unaltered during the updating phase of the Public Symbols. If the check-box is not selected with a tick, it will be updated with those of the reference public symbol's

The reference public symbol will be the one contained in the reference Screen which can be selected through the **"Public Source Container"** property from the **'Screen Execution Properties'**.

## Style Sources in Symbols

The Style Sources in Symbols consent you to activate a certain graphical style for objects in screens based on one style reference (source). In this way, independently from the style associated to objects in design mode, these objects can inherit a style, defined by the programmer, during runtime. The Style Source Container can be activated by the user logged in, from the active language or from a style source screen. This management therefore allows certain graphical aspects of screens and their object contents to be changed during the project runtime mode. The properties which can be edited are:

- **background color**
- **text and edge color**
- **font**

The style sources are acquired by the screen and by the objects it contains and acts as a style container which is not usually displayed in runtime principally because it is only a source container. The controls inserted in this screen work as stile generators. Each object from the style source screen is a style source for the objects belonging to the same category. Only the first control in tabulation order for each object type will be examined and become the style source. For example, when inserting many display objects in the style container screen, only the first tabulation order will act as the style reference. Some objects, even though graphically different, belong to the same category, therefore inherit all the styles of the same object source.

The table below shows which categories the objects are grouped in:

Reference Category	Object
basic	<p>"Basic Shapes" Category:</p> <ul style="list-style-type: none"> <li>• Line</li> <li>• Rectangle</li> <li>• Rounded Circle</li> <li>• Ellipse</li> <li>• Arc</li> <li>• Chord</li> <li>• Pie</li> <li>• Text</li> <li>• Square</li> <li>• Circle</li> </ul>

poly	<p>"Basic Shapes" Category:</p> <ul style="list-style-type: none"> <li>• Polygon</li> <li>• Polyline</li> <li>• Pipe</li> <li>• Polybezier</li> </ul>
Button	<p>"Buttons-Lights-Switches" Category:</p> <ul style="list-style-type: none"> <li>• All the objects</li> </ul> <p>"Advanced shapes":</p> <ul style="list-style-type: none"> <li>• Group</li> </ul>
Gauge	<p>"Sliders-Gauges-Meters-Displays":</p> <ul style="list-style-type: none"> <li>• All the objects apart from the "Editbox-Display" object</li> </ul>
Trend	<p>"Trends-Charts-Data Analysis" Category:</p> <ul style="list-style-type: none"> <li>• Horiz. Plotter</li> <li>• Vert. Plotter</li> <li>• Horiz. Trend</li> <li>• Vert. Trend</li> <li>• XY Trend</li> <li>• Data Analysis</li> </ul>
Chart	<p>"Trends-Charts-Data Analysis" Category:</p> <ul style="list-style-type: none"> <li>• Chart</li> </ul>
GridBox	<p>"Advanced Shapes" Category:</p> <ul style="list-style-type: none"> <li>• Grid</li> </ul>
ListBox	<p>"Advanced Shapes" Category</p> <ul style="list-style-type: none"> <li>• Listbox</li> </ul>
EmbView	<p>"Advanced Shapes" Category:</p> <ul style="list-style-type: none"> <li>• Embedded Screen</li> </ul>
Edit	<p>"Advanced Shapes" Category:</p> <ul style="list-style-type: none"> <li>• Horizontal Spin</li> <li>• Vertical Spin</li> <li>• Editbox-Display</li> <li>• Combo Box</li> </ul>
TabObj	<p>"Advanced Shapes" Category</p> <ul style="list-style-type: none"> <li>• Tab Group</li> </ul>
AlarmWnd	<p>"Advanced Shapes" Category:</p> <ul style="list-style-type: none"> <li>• Alarm Window</li> </ul>

HisLogWnd	"Advanced Shapes" Category: <ul style="list-style-type: none"><li>• Log Window</li></ul>
DLRWnd	"Advanced Shapes" Category: <ul style="list-style-type: none"><li>• DataLogger/Recipe Window</li></ul>
TraceDBWnd	"Advanced Shapes" Category: <ul style="list-style-type: none"><li>• TraceDB window</li></ul>
HourSelector	"Advanced Shapes" Category: <ul style="list-style-type: none"><li>• Scheduler Window</li></ul>
IPCamera	"Advanced Shapes" Category: <ul style="list-style-type: none"><li>• IP Camera Viewer</li></ul>



Style updating is only done in the Runtime phase. therefore the symbols whose styles are edited will show with their original properties during design mode. Movicon will only apply the modifications during the Runtime phase.

## Using Style References

---

In order to manage the style references in symbols you will need to follow the below four points when in design mode:

1. create one or more style source screens for placing shape references in
2. the properties of style source references ("background Color", "Edge-Text Color ", "Text Font") and screen ("Back Color") are set as required. These properties are those which will be propagated as style references for the other project symbols
3. the style source screen name must be inserted in the screen, user, user group or string table language column's "Style Source Container" property. By doing this, the style source of the resource active at that moment will activate according to the priorities described below
4. the properties of the project objects (shapes) ("Style Source Back Color", Style Source Edge-Text Color" Style Source Font) and any of the screens ("Back Color") will be enabled based on the fact that these objects or screens have got to inherit the styles in question from the style source screen

For instance the three properties below are available for each object that can be inserted on screen. When these properties are enabled, they will consent the object to inherit the style from the style source screen:

- Style Source Back Color
- Style Source Edge-Text Color
- Style Source Font

However, a screen can only inherit a back color style due to the one property available to enable this:

- Style Source Back Color

## Activating Style Sources

---

When the style container source is active in runtime mode all screens loaded will be controlled and any object with enabled style properties will be modified to match the styles set in the source object. The Style container source can be activated in different modes. The style source activation procedure, in order of priority, is as follows: when a screen is

loaded it is controlled for any style source container screen associations to the active user. If there are no associations the same control is carried out in the user group which the active user belongs to. If not found another control will take place in the current language being used for one. If this results futile, a final control will be made to see if a style source container screen has been associated to the loaded screen, itself.

The association of the style source container to users, a user groups, languages or screens is carried out through the **"Style Container Source"** available in each one of these resources.

When more than one resource has been associated with a Style Resource Container screen, the active style source will be considered by the priorities described above. If, for instance, an user is active and has been associated a style resource container screen, this screen will be used as the style's source. When the user logs off, the style source container screen will then be used by the active language if enabled to do so otherwise it will be used by the active screen if enabled to do so. If no other style resource screen is activated, the last style loaded will remain active.

The style source is acquired by screens and the objects they contain not only in the screen loading phase but also after they have been loaded. Therefore, when opening a screen its objects may have one a certain graphical aspect which may change to another type if a user, who has been associated with a different style source container screen, logs in.



Once the project is in runtime mode the styles of the objects in the style source container window can be modified using the "DrawCmdTarget" basic script functions. By doing this, all the modifications will be inherited by the objects which used the style source container window in the project.



Associating the screen itself as its "Style Source Container" in the project design phase will generate the following error in the log file:  
"Cannot find the Style Symbols resource container 'Screen1'"



Only the colors from the back ("Back" property) and edge-text (Edge-Text property) are inherited. The colors inherent to other properties, such as the Alarm Window Log colors or the Gauge Warning Zone, etc are not inherited by the styles management.



# 10. ActiveX Object

---

***Movicon fully supports the Active X, OCX and OLE2 standards to allow you to integrate external applications into your project, whether as objects or as Basic Script support for event, method and property management of custom software components.***

Custom software components can be inserted within Movicon projects and used according to the principles and methods of the Microsoft OLE2, ActiveX and OCX standards.

## ActiveX Objects

---

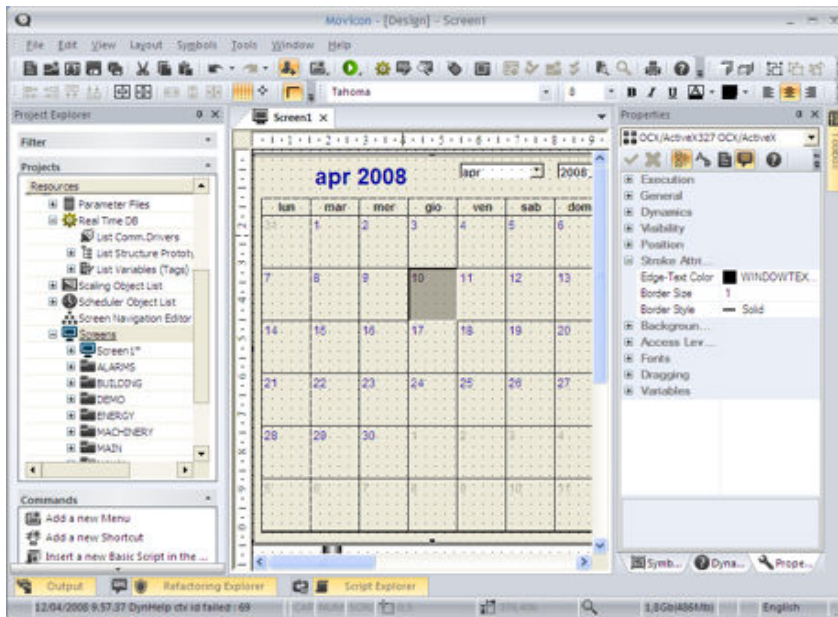
Movicon completely supports the new Microsoft ActiveX standard which allows response to events generated from objects which can be inserted in Movicon screens that have been created externally and not from within the system. This standard, known as ActiveX or OCX, allows you to enhance the operating possibilities of an Movicon application enormously, by letting the integration of any object created by third parties with any compiler or environment which supports this technology.

The ActiveX technology can be considered as an evolution of the OLE technology in the sense that an OLE object can be hosted, used and commanded by a container application which imparts orders and requests. An ActiveX object, however, in addition to this can take actions and generate events which the container application can respond to.

In general, the ActiveX objects are applications with .OCX extensions created by third parties based on Microsoft COM technology (Component Object Model).



Be able to dispose objects independently from applications, has its enormous advantages in terms of the re-usability of work carried out. In addition to this, there are companies who specialize in supplying specific objects as solutions to specific problems. This also adds a new element in the way of producing software. In the future it will be possible to get ready-made objects for inserting into personal applications, or sell parts of applications as ActiveX objects, independently from the language being used or from the type of application which the object is to be used by.



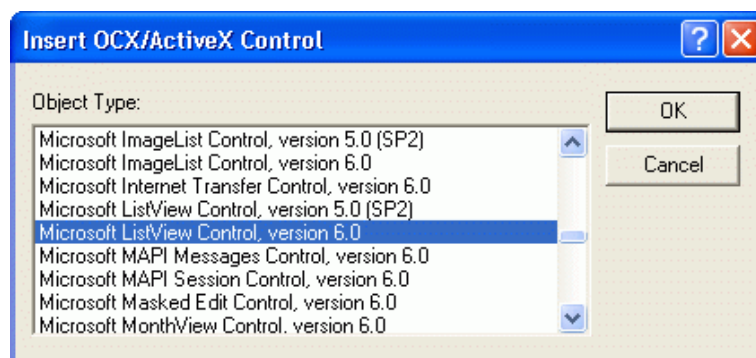
This illustration shows a screen containing the "Calendar Control 8.0 from Microsoft Access™" ActiveX object. This object provides its properties and methods to the application containing it (Movicon) through the window illustrated.

## Inserting an ActiveX

An ActiveX object can be inserted into a Movicon screen by using the **ActiveX** command from the "Special Objects" in the "**Objects Window**". The command will activate a standard window for selecting the object type to be inserted.



The list of ActiveX object types varies according to the applications installed on the PC which support this standard or by the objects installed and recorded in the Windows register.



Movicon will propose an object, such as a Control, to be inserted for default during the insertion phase.

During the insertion phase the ActiveX object can be inserted as a new object created by the application of origin or can be built from the contents of the specified file by selecting the file's Create option.

The Add Control button allows new ActiveX controls to be inserted and registered in the operating system. In this case, the files of origin need to be selected by means of the standard file selection window.

When an ActiveX object is inserted into a screen, it is in fact inserted in the ActiveX object container. This container object is to be considered as a simple rectangle object. All the animation properties, available in the Movicon "Property Windows" when the ActiveX object is selected, all refer to this container.

## ActiveX Editing

---

An ActiveX object inserted into a Movicon screen can be sized as pleased by dragging its borders with the mouse or dragged to any position in the screen area by the mouse. Any changes made to ActiveX/OCX objects are based on how they have been predisposed by those who created them. Each object therefore can provide property setting windows to setup their features. Their features are generally style, graphics which can vary a lot from one object to the next, according to the purpose for which it was designed for. To access the object's settings, you need to use the ActiveX button in the Movicon "Properties Window".



**TAKE NOTE:** It is not always possible to display the object's properties window when inserting an ActiveX. This is often due to incompatibility reasons as ActiveX are sometimes created in Visual Basic, whereas Movicon is created in C++. Movicon will make request to display the ActiveX properties, but if the ActiveX does not interpret this request correctly the window will not display. As a remedy, the ActiveX properties can always be accessed through its Basic Script functions. By means of using the "Script Explorer" window you can access the ActiveX's edit code window, where its events, functions and properties are found.

## Editing ActiveX using VBA

---

ActiveX controls can also be edited on event in function with the properties provided by the control itself. The same object's method and property browser can be activated from the object's "Script Explorer". Based on how the object has been created, the code can set the properties according to the normal VBA™ comp. programming procedures described in the section dedicated to the Movicon VBA programming. As previously mentioned, it is not often possible to change ActiveX object properties through the Properties Window, therefore the object must be configured by using Basic Script codes.



**Note! The ActiveX is entered as 'ActiveXInterface'. Therefore the methods and properties are called by using this interface (ActiveXInterface.<property or method>). The ActiveX events are also available from within the 'ActiveXInterface' interface.**

For instance, let's presume that we have to access the "Day" property of an ActiveX "Calendar" from the object's script code we would then have to write:

```
Option Explicit
Public Sub ActiveXInterface_Click()
    Debug.Print ActiveXInterface.Day
End Sub
```

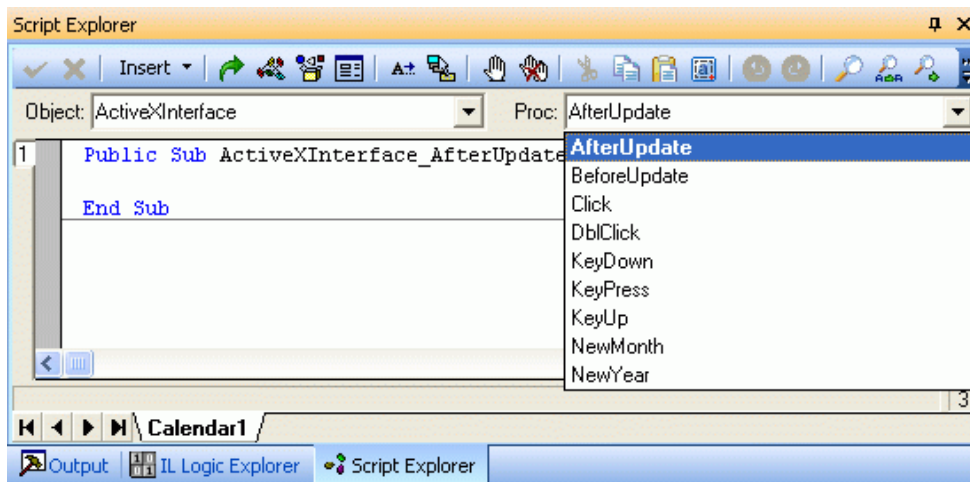
### ActiveX object event management

---

The purpose of an ActiveX object inserted in a Movicon screen is to create events in function with the methods predisposed in the object itself. The Movicon programmer can insert Basic Script codes (VBA™ comp.) to execute the application's management and



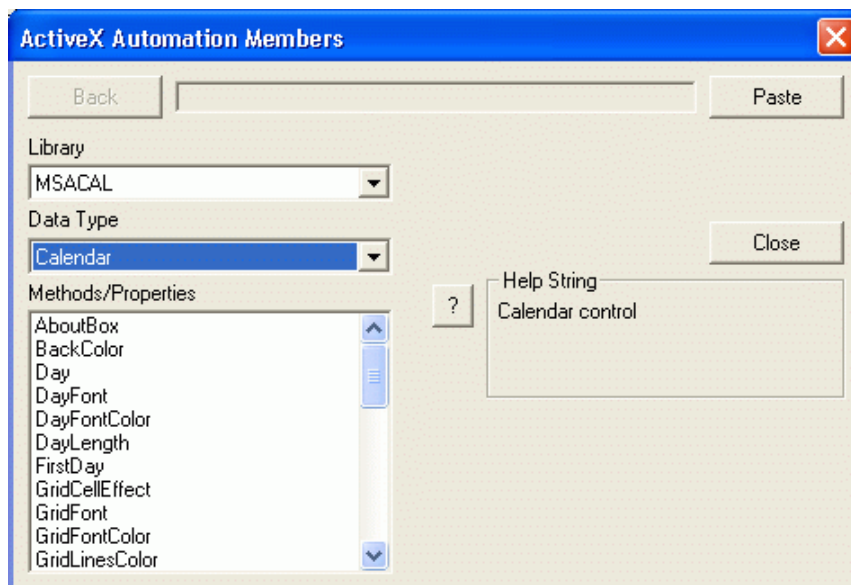
control logic to verify the event desired, provided by the ActiveX control. Code editing is done through the "Script Explorer" window after the object has been selected:



By means of the "Script Explorer" window you can select the event (or procedure) from those provided by the object. In order to do this you must first select the **"ActiveXInterface"** item from the list box called **"Object"**: and then select the desired event from the list box call **"Proc."**. The code can be then entered and will be executed in runtime when the ActiveX generated the event.



The events provided are determined by how the object has been predisposed by those who created it. For further information on the operative modalities of the Basic Script codes, please consult the appropriate section dedicated to programming Movicon Basic Scripts.



In response to the events you can also change the object according to the properties and methods provided by the object's creator.

ActiveX object properties and methods, if available, can be viewed by using the function browser. The browser is activated by using the Browse command found in the "Script Explorer's" tool bar.

## How to reference an Active from a screen's symbol

It is often found necessary to give commands to the ActiveX object from objects contained on screen, such as buttons. In order to do this you need to create an object

from the button's code to refer to the ActiveX. It will then be possible to use the methods and properties of that ActiveX through this object. In order to do this you will need to use some of the specific Basic Script functions. The example below shows you the necessary steps to take:

Example: Let's suppose that a Calendar ActiveX type has been inserted on screen in which a button is to be used for displaying the days selected in the Calendar. The basic code for this button would be:

```
Option Explicit
Public Sub Click()
    Dim objContainer As Object
    Dim objCalendar As Object

    Set objContainer = GetSynopticObject.GetSubObject("Calendar1")
    Set objCalendar = objContainer.GetActiveXObject
    ' properties depend on the object type
    MsgBox "Selected Day = " & CStr(objCalendar.Day), vbInformation,
    GetProjectTitle
    Set objContainer = Nothing
    Set objCalendar = Nothing
End Sub
```

The "objCalendar" object is the Calendar object, therefore provides all the methods and properties belonging to the ActiveX.

## Active X properties

---

The ActiveX object can receive some properties, including those of animation, independently from the object type and which can be set through the Movicon "Properties Window". A major part of these properties refer to the Active X object container, and are all generic properties available to each Movicon control.

### Active X Execution Properties

---

The ActiveX execution properties allow you to manage its the license. The Active X objects have always had a license which may come free of charge or must be purchased according to each certain case. The licenses of some ActiveX objects, or ActiveX objects themselves, are installed with a developer software packet, such as MS Office2000/XP, Visual Basic, etc.

To change the Execution settings of an ActiveX object, select the object with the mouse and use the Movicon "**Properties Window**".

#### License

After the Active X license code has been inserted with the "**Get License**" command, it is displayed in this box.

#### Get License

This command allows you to get the license code of the selected ActiveX. When the command does not return any code, this means that the ActiveX license has not been installed and therefore the ActiveX will not be able to work. In certain cases the ActiveX Runtime license may be inserted only. This happens for some Microsoft ActiveX which are installed with the Runtime only license with the operating system. The development license, however, is installed with supplementary software such as MS Office2000/XP, Visual Basic, etc.. In this circumstances the project can be executed in Runtime, but no editing can be done to the ActiveX object.

## ActiveX General Properties

---

The ActiveX General Properties allow you to open the ActiveX Properties Window. All the other properties of this group are used for configuring the ActiveX container object and are the same properties available to any other Movicon object.

To edit the ActiveX General settings, select the object with the mouse and use the Movicon **"Properties Window"**.

### ActiveX Properties

This command is used for opening the ActiveX properties window.



**TAKE NOTE:** It is not always possible to display the object' properties window when inserting an ActiveX. This is often due to incompatibility reasons as ActiveX are sometimes created in Visual Basic, whereas Movicon is created in C++. Movicon will make request to display the ActiveX properties, but if the ActiveX does not interpret this request correctly the window will not display. As a remedy, the ActiveX properties can always be accessed through its Basic Script functions. By means of using the "Script Explorer" window you can access the ActiveX's edit code window, where its events, functions and properties are found.

## OLE Objects

---

Movicon completely supports the Microsoft standard for inserting objects into the project which come from other dynamically linked applications.

This standard, largely diffused in applications for Windows, is called O.L.E. (Object Linking and Embedded vers.2.x).

The OLE is a unification of services in the form of objects which allow customized software architectures to be created by deeply integrating software components of different applications. In short, the OLE offers a coherent standard that allows objects and applications to communicate with one another by using each others code.



Movicon is a OLE2 container application and a OLE2 Automation Server application.

The OLE objects can be inserted into any Movicon screen.



A typical example of OLE object usage would be to build on a EXCEL™ spreadsheet inserted into a Movicon screen. This inserted object would be linked to the Microsoft EXCEL™ application to work as it normally would even though inserted in a Movicon project.

The type of OLE objects, which can be inserted into Movicon projects, depend on whether the applications installed on the hardware platform are capable of supporting this standard.

## Inserting OLE objects

---

OLE objects can be inserted into Movicon screens by using the **OLE** command found in the "Special Objects" group from the **"Objects Window"**. The command activates a standard window for selecting the application type from which the object, to be inserted, is taken from.



The list of OLE objects varies according to the applications installed on the PC which support the OLE standard.

The OLE objects can be inserted into any Movicon screen.



The OLE object which can be inserted in to Movicon projects depend on the applications installed on the hardware platform capable of supporting this standard. During the insertion phase the OLE object can be a new object created from the application of origin or it can be built by the contents of the specified file by selecting this file's create option.

## Editing OLE objects

Once inserted into a Movicon Screen, the OLE object can be sized as pleased by dragging its borders with the mouse or dragged to any position of the screen area.

The OLE objects are edited by means of **using the verbs**, these are purpose-built commands from the application of the inserted OLE object (these verbs are usually **Edit**, **Open** and **Convert**, but may vary from one application to the next).

The verbs for executing commands or manoeuvres on the object are always available in the programming mode by using the command referring to the Object from the Edit menu or by using the right mouse key with the OLE object selected.

The OLE object's edit command can also be executed straight away by double-clicking the left mouse key on the object.



During runtime mode, the use of the verbs (Open, Edit,...) for executing commands or manoeuvres on the object is established by the programmer, in function with how the properties of the OLE object were configured in merit of using verbs.

The commands related to editing the object bring about a change in the functioning of the Movicon menu. The menu actually adapts its items by inserting the command itself of the application deriving from the inserted OLE object.

This all makes it possible to work with the object as if in the object's own application, but within the Movicon environment.



Since there are so many applications which support the OLE and OLE2 standards, it is impossible to describe all the relative commands and options. Therefore you should refer to the manuals of the respective applications for get details on how these operative commands and options are used in the objects.



# 11. Menu

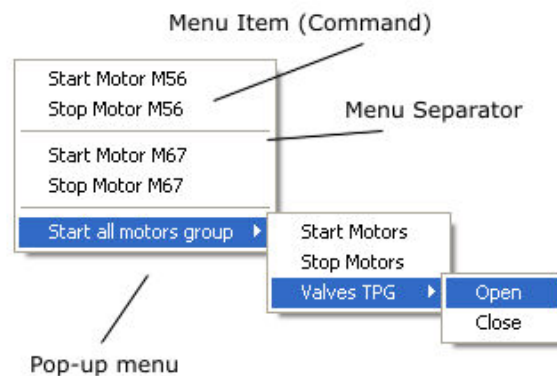
***The Movicon graphic interface permits pop-up Menus to be created and activated with the mouse, or menu bars to be customized and associated to the video screen pages.***

The Movicon Menus are very useful in projects for creating a complete man-machine interface.



Pop-up Menus can be created with Movicon which can be organized in sub-menus in a tree structure, or created as customized menu bars in screen pages.

By means of the Movicon Menus you can set commands within the project or the plant, and verify if their execution took place with a check sign (✓) displayed at the side.



The above figure illustrates an example of a pop-up sub menu structure. The figure below shows the relating tree structure in the project.

Resources	Command
[-] List Child Projects	
[-] Menu	
[-] Motor Commands*	
[-] Motor Commands	
[-] Command All	-
[-] Motor 1 Start	
[-] Motor 1 Stop	
[-] ---	
[-] Motor 2 Stop	
[-] Motor 2 Stop	
[-] Valves Commands	
[-] EV2	
[-] Close	
[-] Open	
[-] EV1	-

The Menu resource offers numerous operating options in any application. For example, you can activate as many menus as there are as many actuators in a Screen page, which can be activated by means of 'Hot Regions' for setting manual ON-OFF commands on the same actuators. You can also display a guide string to each menu item on the Movicon Status Bar.

## Inserting Menus

---

When inserting a Menu in the project the procedures for inserting new objects in the "Menu" group in the 'Project Explorer' window have to be carried out. This can be done by either right mouse clicking on the "Menu" group in the 'Project Explorer' window and then selecting the 'New Menu' command, or by using the purpose built icon in the Movicon toolbar. When using the icon keep the right mouse button pressed on it for about a second to open a pop-up window where you can select the resource to be inserted being the Menu resource in this case.

When confirming the operation the new menu will appear in the group or the point selected in the project's tree structure. From this point the procedures for setting the menu's properties can be carried out as described in the document about the **"Menu Properties"**.

The Menu resource can subsequently be assigned a Name by either clicking the resource and entering the name replacing the one for default, or select the resource and press the F2 key and proceed with inserting a new name.

### Importing Menus from other Projects

---

Movicon allows one or more Menus to be copied from one project to another. In order to do this you must first open both projects, then select the menu from the source project's Project Explorer window, execute the Copy command, then position the mouse pointer in the destination project's "Menu" group of the Project Explorer window and execute the Paste command. The copied Menus will then also be available in the destination project.

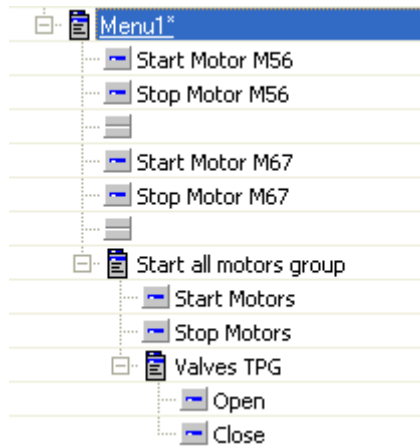
The Drag & Drop techniques can also be used with the following procedure: select the Menu from the source project and by keeping the left mouse key pressed drag it to the point desired in the destination project, then release the mouse key.

## Menu Items

---

Each project menu is composed of one or more Items, which are components belonging to the menu. The Items are displayed in a tree structure within the **'Project Explorer'** window.

From this window the items, which are to compose the menu, can be inserted, configured and be structured as sub-menus. To add new item use the right mouse button or the **"Command"** window from the **'Project Explorer'**.



There are three types of items used for composing menus:

- **Normal:** menu items through which operating commands are set
- **Separator:** menu items which represent separator lines between each menu item. These items are only graphical and do not have any execution properties
- **Pop-up:** menu items which branch off items in submenus. These items are only used for structural purposes and do not have execution properties.

The right mouse button is used for inserting item into menus after having selected the Menu or the position required, or by using the **"Command"** window from the **'Project Explorer'**.

To edit or modify the items use the Movicon **'Properties Window'**.

To delete one or more items from the menu, select and activate the Delete command by using the DEL key or the 'Edit' system menu.

Any editing can be cancelled or restored with the **Undo** and **Redo** commands.

The techniques used for working on a menu structure are equivalent to the ones used for project tree structures.

## Change Item Order

Items are inserted by Movicon at the end of those already on the menu. You can however change the order of these items by using the **"Change Menu Item Order..."** command which opens the following dialog window:



To change the Item order just select, drag and drop them with the mouse in the position desired.



## Test Menus

---

Tests can be carried out on menus during the editing phase to check the menu's graphical outcome straight away.

The 'Test Menu' command can be accessed by using the right mouse key or the **"Command"** window from the **'Project Explorer'** to immediately display the Items as they will appear in the Menu during Movicon Runtime.



If a menu has been assigned with the same name as a Screen, it will be displayed as a **'Menu Bar'** in the window in question during runtime mode.

## Customizing Menu Bars

---

Movicon allows you to create custom **"Menu Bars"** to associate to video screen pages. To associate the Movicon bar menu to a certain Screen simply assign the menu required (created with the normal creating menu procedures) with the same name of the Screen through which the Menu bar is to be displayed.



**The menu associated to the "Startup Screen" will be displayed as a default menu for all the other Screens unless menu resources with the same Screen names already exist.**



**In cases where the project has not been defined a "Startup Screen" but has a "Main" menu, this menu will be automatically loaded at the project startup instead.**

## Menu Properties

---

The menus inserted in the 'Project Explorer' window's **"Menu"** folder can be completely customized in the properties. In order to do this just select the Menu or Item required and then edit their settings by using the Movicon **'Properties Window'**.

### Menu General Properties

---

The General properties are used for setting the positions where the Pop-up menu is to appear. In order to do this just simply select the Menu required and then edit its settings through the Movicon **'Properties Window'**.

#### **Name**

This edit box permits you to define the Menu object's name which is to be configured.

#### **X Pos**

This edit box is used for entering the X coordinates in pixels of the position where the Menu window is to appear within the workspace. The coordinates refer to the position of the Menu's top left corner in respect to the screen's top left corner (being X0).

To make the menu appear where the mouse pointer is positioned, set (or leave set) the default value at "-1".

The menu's coordinates in the workspace can be checked out by carrying out a 'Test Menu' with the appropriate command.

#### **Y Pos**

This edit box is used for entering the Y coordinates in pixels of the position where the Menu window is to appear in the workspace. The coordinates refer to the Menu's top left corner in respect to the screen's top left corner (being Y0).

To make the menu appear where the mouse pointer is positioned, set (or leave set) the default value at "-1".

The menu's coordinates in the workspace can be checked out by carrying out a 'Menu Test' with the appropriate command.

## Item Menu General Properties

---

The Menu Item's main features are setup through its General properties. In order to do this simply select the Item required and edit the settings through the Movicon **'Properties Window'**.

### Type

This option box is used for defining the Item type to be included in the menu. The Item's characteristics have to be set according to the type of Item selected as described below:

- **Normal:** activating this selection will display the item as a normal menu item. The item has to be associated with a text description which will appear as the menu item, to which a command execution will be associated from the **"Menu Item Execution Properties"**
- **Pop-up:** activating this option, means that the menu item is to display a further list of items. The 'pop-up item will be marked by a "▶" symbol and if activated will display the items appropriately inserted in the menu's tree structure.
- **Separator:** activating this option means that line separator will be displayed between each menu item. A separator is usually used when a division between two items needs to be highlighted. A separator item can not be executed and therefore does not have any execution properties.

The number of items (pop-up, normal or separator) that can be inserted into each menu is virtually unlimited.

### Caption

The text used as the item's title is entered in this edit box. The text will be displayed in the menu as an item which can be associated with an execution command or a pop-up submenu.

The caption is not available when the separator item is being used.

### Prompt

The text string, which is displayed in the status bar when the menu item is selected is entered in this edit box.



The status bar, (found at the bottom of the workspace) can serve as an information guide for the Movicon menu items.

### Image

This property is used for selecting the image to be displayed in menu in the place of the title.

### User Level

The Password User Level is entered in this property which is needed for executing, for instance, the command lists associated to the Menu Item. When the Menu in question is to be used by a user, Movicon will request activation of a user who has a Password User Level equal to or higher to that set in the Item itself. If the user, logging on, has the necessary access rights they will be given correct authentication, otherwise they will be denied Logon and it will not be possible to execute the operations requested.

For further information see paragraph **"User Level and Access Level"**.

## Menu Item Execution Properties

---

By using the Item Execution properties you can associate the required command to be evoked when it is selected and activated from the menu. In order to do this just select the item required and then edit its settings through the Movicon **'Properties Window'**.

### Enable

This edit box is used for inserting the name of the variable (or select it with the '...' browse button on the right) which will activate the item from the menu. When the variable obtains the 'zero' value, the menu item will appear active and then available for command execution. When nothing has been specified in this box by being left empty, Movicon will consider this item as being enabled.

### Check

The name of the variable which determines the displaying of the tick or check sign (✓) at the side of the menu item (or select it with the '...' browse button on the right). If nothing is specified in this option, the check sign will not display.



The check can be used for indicating the status of the command associated to the item.

### Commands

This button opens the Movicon **'Command List'** window where a list of one or more commands, which are to be executed when the Menu Item is activated, if defined. For further information on the available commands please consult the **"Command List"** paragraph.

## Menu Example

---

Let's suppose you want to configure your project so that when clicking with the mouse on an area in the Screen representing a pump called 'P1', a menu is displayed indicating the 'Manual' and 'Automatic' items.

The 'Manual' item must contain the 'ON' and 'OFF' items.

First of all the menu has to be edited (which will be called 'P1 Pump menu').

Then we will proceed to call up the menu by using the Hot Region control purposely located in the Screen.

Proceed with editing the Movicon Menu as follows:

1. Create a new Menu Resource from the 'Project Explorer' window
2. Then execute the command for inserting a new Item into the menu
3. The menu's new item has to be configured to contain the 'Manual' text title and set as 'Pop-up' type
4. When you have configured the Pop-up 'Manual' item, you can then insert a new item which will branch off from the previous one. Configure the new item as 'normal' type and assign the 'ON' title (with execution command "Set" at '1' value for the start pump variable)
5. Insert a new 'normal' item which will belong to the same branch. Configure the new item and assign the 'OFF' (with execution command "Set" at '0' value for the start pump variable)
6. At this point select the icon representing the root of the menu with the mouse (usually called 'Menu1' by the system) then insert a new 'normal' item which will be added underneath the 'Manual' item in the menu. Assign the 'Automatic' title to the new Item and specify the command type to be executed (eg. set the automatic cycle to value '1')
7. To verify the menu just edited, use the 'Test Menu' command. The menu will display exactly as it will when activated in Runtime mode
8. The new menu will be added to the resource's tree structure still with the provisional name assigned by the system (usually 'menu1')
9. Select the new menu and assign it with the name 'Pump P1 Menu' by means of the 'Properties Window'

You have now completed the menu and can now activate the Screen resource representing the pump P1 drawing.

Insert the Hot Region object, after which you can position and size it as required then configure it by assigning it with the Menu activation command in the execution properties.

When specifying this command you can select the menu you have just edited and named 'Pump P1 Menu' from the 'Menu' list.

Run the project and click on the pump P1 drawing to display the 'Pump P1 Menu'.

# 12. Shortcuts

---

***The Movicon graphical interface permits complete and easy keyboard usage, by associating commands to keys or combination keys in function with the active Screen.***

This section describes how to use and set the Movicon Shortcut Resources in projects. The Movicon Shortcuts (or keyboard commands) can be extremely useful in projects for using the PC keyboard to execute commands on the plant or operational commands in the project itself.

Each Movicon Shortcut resource can be considered as a set composed of one or more keyboard commands.

Each command set is obligatorily associated to an Screen by Movicon, therefore allowing command sets (Shortcuts) to be assigned in function with the active screen page.



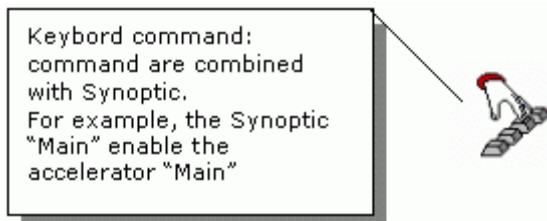
The name assigned to each Shortcut Resource is very important as the Shortcuts are associated to their Screens according to their name. The Shortcut must have the same name as the Screen's in which they are to be active.



The Shortcut associated to the "Startup Screen" will automatically be activated by the system upon starting up the project. In addition to this the Shortcut associated to the "Startup" will also be activated each time a Screen, which has not been associated any Shortcuts, is loaded.

By means of the Movicon Shortcuts keys or combination keys can be used for executing commands on the plant, for setting variables or executing any command type included in the **"Command List"** offered by Movicon.

Using keyboard commands are essentially necessary in plants where the mouse or other such devices cannot be used.



The Movicon Shortcut keyboard commands, when active, always have top priority over the same keys or combination keys used by Windows for operating system commands as the Shortcut provides command activation upon being pressed and not released.

Example: If the Windows system provides the F1 key for activating the guide and the Movicon Shortcut is active which has the same command for executing upon pressing the F1 key, the command associated in the Shortcut will be given priority.

# Shortcut Activation

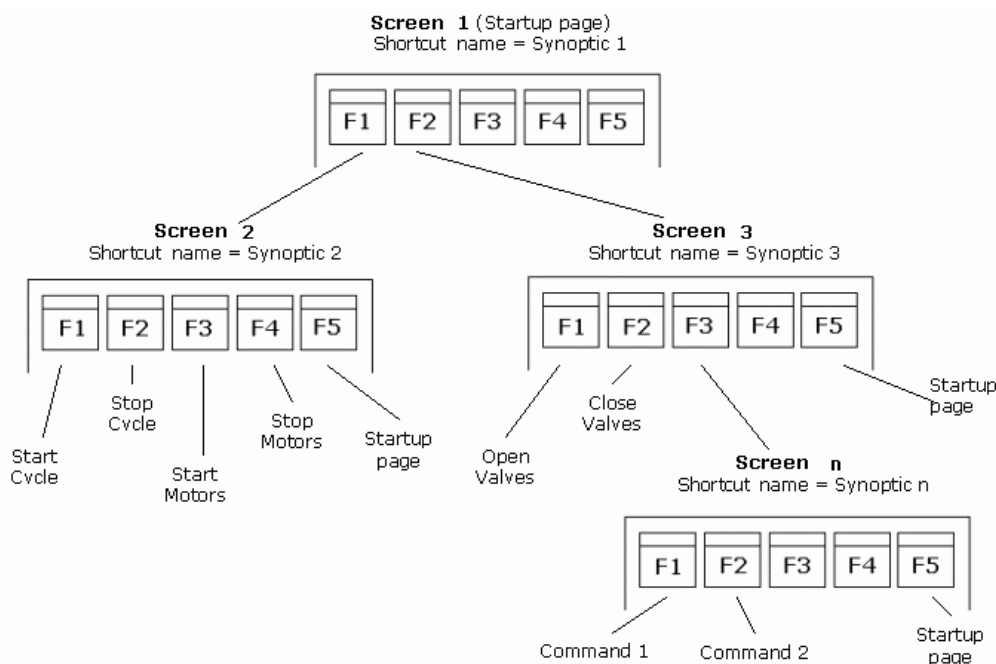
Movicon is predisposed in order that an Shortcut associated to the **"Startup Screen"** is activated at its start up together with the "Startup Screen". If there is no Shortcut associated to the "Startup Screen", no Shortcut will be activated for the present time. When a Screen is activated in the project in Runtime Mode, Movicon will also check whether there is an Shortcut resource with the same name as the active Screen. If one is found it will also be activated otherwise the main Shortcut will be activated for default being the one associated to the "Startup Screen". When the Shortcut associated to the "Startup Screen" is not present no other will be activated until a Screen, with one associated to it, is opened.



The Screen Windows and the Shortcuts are therefore closely related.

The association of Shortcuts to Screens permit the same keys to be associated to different functions relating to the screen page displayed.

This illustration shows an example of how Screens and Shortcuts are related:



Relation between Screens and Shortcuts. A set of keyboard commands is always disposed in relation to a screen page or Screens.

## Inserting Shortcuts

In order to insert Shortcuts in to the project you need to carry out the procedures for inserting a new object into the "Shortcut" group in the 'Project Explorer' window. Inserting a new Shortcut can be done by right clicking the mouse on the on the "Shortcut" group in the 'Project Explorer' window and selecting the 'New Shortcut' command, or by using the appropriate icon from the Movicon tool bar. In this case by keeping the left mouse button pressed for about a second a drop-down window will display where the resource to be inserted can be selected.

The new Shortcut will appear in the group on the point selected in the project structure when the operation is confirmed. At this stage you can proceed with setting the Shortcut's properties as described in the documents about **"Inserting Shortcuts"**.

A name can then be assigned to the Shortcut resource replacing the one proposed for default or after having selected the resource press the F2 key and proceed with entering a new name.

## Importing Shortcuts from other Projects

---

Movicon allows the copying of one or more Shortcuts from one project to another. To carry out this procedure both projects need to be opened first, then after having selected the Shortcuts from the Project Explorer window of the source project execute the Copy command. Next position cursor in the "Shortcut" group of the project Explorer window of the destination project and execute the Paste command. The copied Shortcuts should then be available also in the destination project.

The Drag & Drop technique can also be used with the following procedure: select the Shortcut from the source project and keep the left mouse key pressed drag it to the point desired in the destination project then release.

## Shortcut Keyboard Commands

---

Each project Shortcut consists of one or more keyboard commands activated by the relevant associated keys.

In addition to an activation key the virtually unlimited commands that can be inserted in each single Shortcut resource can be associated to keys called 'Modifiers', being those key Combinations multiplying the operating possibilities on the keyboard. The Modifiers keys are ALT, CTRL and SHIFT.

The Shortcut commands can be viewed through the **"Command List"** window, which can be accessed from the **'Shortcut Execution Properties'** window. This window is used to enter and configure the Shortcut commands.

New keyboard commands can be entered by using the 'New Shortcut Command' when selecting an Shortcut by means of the 'Project Explorer' window, which is made available by using the right mouse key or the 'Commands Window'.

Select and activate the CANCEL button to delete one or more commands from the Shortcut.

The Shortcut command properties can be set by using the Movicon 'Properties Window'.



Identical key combinations associated with different commands will execute the first command taken into consideration.

Any active Movicon Shortcut keyboard commands will always have priority over the same keys or combination keys used by Windows for system operating commands, as the Shortcut provides command activation upon pressing the key and not upon releasing it.

# Speech Recognition

---

Movicon provides a very handy feature which is a function that recognises spoken texts associated to the Shortcuts resource for activating commands vocally.



**Only the "Microsoft English Recognizer v5.1" speech recognition engine is available at the moment. The Speech Recognition function can not be used for the Italian language.**

In order to use the speech recognition function for activating commands you need to first check whether your PC has been fitted with a microphone for giving vocal commands. Your PC will also need an audio card to allow the system to enounce request and confirm messages.

Having done this, you will need to verify that you have Windows SAP 5.1 and that it is configured correctly. This can be done by going to 'Speech' found on the control panel. We strongly advice you follow the guidelines on microphone configurations procedures and speech profile regulations.

At this point, enter the Movicon development environment and add a new shortcut resource using the Project Explorer window. Then go to its "Shortcut General Properties" and "Shortcut Command Execution Properties" to define the parameters for defining speech command recognitions. Once the configurations have been completed and the Shortcuts activated for managing the spoken command recognitions you should be able to activate commands by pressing relative assigned buttons or by simply enouncing the texts assigned to the vocal commands.

The speech recognition can also be managed in multilingual. All the texts that can be associated to a Shortcut's speech recognition properties can be inserted using Sting IDs from the String Table. **However, as mentioned above, the speech recognition engine is only available in English and therefore texts associated in these properties must always be in English no matter what language has been activated in the project.**

## System Variables

During the runtime phase you can use some System Variables for verifying which Shortcut is active and the working status of the speech recognition. These variables, whose meanings are described in the relating "System Variables" section, are as follows:

```
_SysVar_:SRActiveShortcut  
_SysVar_:SRListening  
_SysVar_:SRIsActive  
_SysVar_:SRLastUnderstood
```

## Example:

---

Below you will see how to use the Shortcut properties to configure a command for exiting from the application. Those properties not mentioned can be left set with their default values.

### Shortcut General Properties:

- Don't understand Text: "Don't understand Text"

### Shortcut Command Execution Properties:

- Enable Speech Recognition: Enabled
- Text Command: "Exit"
- Require Confirm: Enabled
- Prompt Confirm: "Confirm"
- Confirm Text: "Yes"

At this point conversation between User and System during project Runtime will be as follows:

- User: "Exit"
- System: "Confirm Exit?"
- User: "Yes"
- At this point the application's exit command will be activated.

## **Speech Recognition for Numeric and Alphanumeric pads**

---

When a Shortcut is associated to a command requiring "Numeric Pad" or "Alphanumeric Pad" to set a variable value and the "Enable Speech Recognition" property is active, the pronunciation of the "Text Command" for Shortcut vocal activation will not open the "Pad" window, but will set the system in listening mode to acquire the value to be set for the variable. At this point just pronounce the value to be set for the variable. Please consider the following notes:

- When a vocal command aims to set a variable value, as if a "Numeric Pad" is used, you should always start the value pronunciation adding "zero", in order to allow the SR (Speech Recognition) engine to convert the vocal text in a number and not in a string.

Examples:

- 'zero one point two' corresponds to the '1.2' value to set
- 'zero five zero' corresponds to the '50' value to set
- 'zero twenty' corresponds to the '20' value to set

- When the command is "Numeric Pad"-like, the pronounced value will not be accepted if out of the ranges defined for the command, or if does not correspond to a number. When the command is "Alphanumeric Pad"-like, the pronounced text will not be accepted if containing a number of chars higher than the one specified in the command. In both cases you can define an "Invalid Value Text" message.
- When a vocal command aims to set a variable value through the "Numeric Pad" or "Alphanumeric Pad", if the "Require Confirm" option has been enabled, the vocal confirmation should be given twice: the first time to confirm the command execution, the second time to confirm the pronounced set value.

### **Example:**

Please find below an example of how to configure a Shortcut properties to obtain a command for a numeric variable value set using a "Numeric Pad". The properties not specified here can be left with the default values:

### **General-Speech recognition Shortcut properties:**

- Don't understand Text: "Don't understand"

### **Execution-Speech Recognition properties of a Shortcut command:**

- Enable Speech Recognition: Enabled
- Text Command: "Numeric"
- Require Confirm: Disabled
- Value Prompt Text: "Insert Value"
- Invalid Value text: "Value not valid"

At Runtime, a possible conversation between the operator and the system could be as follows:

- Operator: "Numeric"
- System: "Insert Value"
- Operator: "zero one point five"
- At this point the value "1.5" will be set for the variable



# Shortcut Properties

---

Shortcuts inserted into the 'Project Explorer' window's **"Shortcuts"** folder can be completely customized in the properties. In order to do this just select the Shortcut or Command required and then edit their settings through the Movicon **'Properties Window'**.

## Shortcut General Properties

---

The Shortcut General Properties are used for setting the Shortcut to be active also during the opening of a modal Screen.

In order to do this just select the Shortcut and then edit its settings through the Movicon **'Properties Window'**.

### **Name**

This edit box permits you to define the Shortcut object's name which is to be configured.

### **Active On Modal**

By enabling this property the Shortcut will be made active also during the opening of a modal Screen. In this case if an Shortcut has been associated to the modal Screen this will be active but if one has not the one associated to the Screen in background will become active. If this does not also have one the startup Shortcut will be made active.

### **Wait Speeching Time**

Interval time in milliseconds in which the system waits for a vocal input.

### **Go to Sleep Command Text**

The word or phrase which the user must pronounce for temporarily deactivating the speech recognition function. The system will reactivate only when the word or phrase entered in the 'Wake Up Command Text' property is pronounced.

### **Go to Sleep Ack. Text**

Word or phrase the system will pronounce, using the PC speakers or a predefined audio output, for confirming temporary deactivation of the speech recognition function.

### **Wake up Command Text**

The word or phrase which the user must pronounce for reactivating the speech recognition function.

### **Wake up Ack. Text**

The word or phase which the system must pronounce to confirm that the speech recognition function has been reactivated.

### **Don't understand Text**

The word and phase which the system pronounces for indicating that it hasn't understood the spoken word/phrase. The word or phase is followed by the word or phase which the system is thought to have heard.

## Shortcut Command General Properties

---

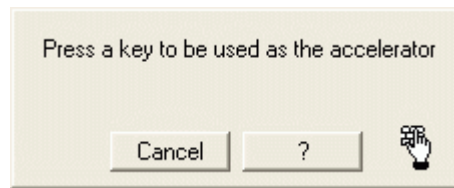
The Shortcut Command General Properties are used to define whether the Shortcut must also be active during the opening of a modal Screen.

In order to do this just select the Shortcut and then edit its settings through the Movicon **'Properties Window'**.

### **Shortcut Key**

This edit box is used for entering the key (or combination keys) to which the command is to be associated to. Movicon offers the possibility to enter the key to be associated by

pressing the key directly from the keyboard. In order to do this press the "... " button on the right of the edit box which will display the following window:



Then just press any one of the keyboard keys or a combination of keys which also includes the 'modifier', as this is recorded in the 'Shortcut Key' box. The operation can be cancelled with the Cancel button.

#### **User Level**

This property is used for setting the Password User Level necessary for executing, for example, the command list associated to the Shortcut command. When the Shortcut in question is to be used by a user, Movicon will request activation of a user with a Password User Level equal or higher than that set in the command itself. When the user executes Logon with correct authentication they will have all the access rights necessary, otherwise if Logon fails and access is denied they will not be able to execute the operations they require.

For further information please refer to the paragraph on **"User Levels and Access Levels"**.

## **Shortcut Command Execution Properties**

---

The Shortcut Execution properties are used to determine whether the Shortcut should also be active or not during the opening of a modal Screen.

In order to do this just select the Shortcut and then edit the settings through Movicon **"Properties Window"**.

#### **On Release**

This property is used to establish whether the command should be activated upon pressing or releasing the key. When the required key in programming stage has already been associated with the Windows standard functions (eg. F1 for the Guide), always leave this property unchecked.

#### **Enable**

The name of the variable which determines the activation of the Shortcut command is entered in this edit box (or selected with the browse '...' button on the right). When the selected variable obtains a 'zero' value, the Shortcut command will not be available. When the selected variable obtains another value apart from 'zero', the Shortcut command will be available for execution. If nothing is specified in this option by leaving it empty, Movicon will consider the Shortcut command as being enabled.

#### **Commands**

The Movicon **'Command List'** window is opened by means of this button, through which a list of one or more commands, to be executed when the Shortcut command is activated, is set. For further information about the commands available please consult the paragraph on **"Command List"**.

#### **Enable Speech Recognition**

Enabling this option will activate the shortcut command speech recognition function.

#### **Text Command**

The word or phrase that the user must say for activating the command (or commands) associated to the Shortcut.

#### **Require Confirm**

Enabling this option will activate the confirm request, by the system, before the command (or commands), associated to the Shortcut, is activated.

**Prompt Confirm**

The word or phrase which the system must say, through the PC's speakers or a predefined audio output, for requesting a confirm to activate the command (or commands) associated to the Shortcut. This word or phrase is followed by the word or phrase entered in the "Text Command" field.

**Confirm Text**

The word or phrase the user must say for confirming the activation of the command (or commands) associated to the Shortcut.

**Value Prompt Text**

The word or phrase which the system says when waiting for input value. This is used in case when the command associated to the Shortcut is "Numeric Pad" or "Alphanumeric Pad" type. When the speech recognition is active, the system will prompt a vocal input instead of displaying the numeric or alphanumeric keypad for receiving inputs from the user.

**Invalid Value Text**

The word or phrase enounced by the system for indicating that the vocal input it received was incorrect (ie. when waiting for a numeric input and the word pronounced is the wrong number).

# Shortcut Example

---

How to configure your project so that the plant operator can view the following screen pages in Movicon Screens:

- Plant Lay-out represented in the "Main" Screen
- Storage Silos represented in a Screen called 'Silos'
- Working Area represented in a Screen called 'Working Area'

Preset the keyboard commands or Shortcuts so that:

the last two above mentioned Screens are called up by using the F1 and F2 keys from the 'Main' Screen

the Lay-out page can be returned to by using the ESC key

in addition to returning back to the 'Working Area' Screen with the ESC key an modal Screen window is activated for setting parameters by pressing the F1 key

Proceed with Editing the Movicon Shortcuts as follows:

1. Create a new Shortcut resource from the 'Project Explorer' window and call it 'Main'. This Shortcut resource will always be activated whenever the 'Main' Screen is loaded (by setting up the 'Main' Screen as the project's startup Screen, the 'Main' Shortcut will consequently become the project's default Shortcut). Edit the new Shortcut by adding the commands for calling up the 'Silos' Screen by pressing the F1 key and calling up the 'Working Area' Screen by pressing the F2 key.
2. Create a new Shortcut resource from the 'Project Explorer' Window and call it 'Silos'. This Shortcut resource will always activate whenever the 'Silos' Screen is loaded. Edit the new Shortcut by adding the command for calling up the 'Main' Screen by pressing the ESC key.
3. Create a new Shortcut resource from the 'Project Explorer' window and call it 'Working Area'. This Shortcut resource will always be activated whenever the 'Working Area' Screen is loaded.
4. Edit the new Shortcut by adding the command for calling up the 'Main' Screen by pressing the ESC key and the command for calling up the modal Screen for setting the plant parameters (the modal Screen must be created beforehand) by pressing the F1 key.

The example project described above can also be set up in another way:

Associate the Shortcut commands so that each key carries out its own function independently of the active Screen.

The following needs to be achieved, as in the example above:

- The F1 key must always call up the 'Silos' Screen
- The F2 key must always call up the 'Working Area' Screen
- The ESC key must always call up the 'Main' Screen
- The F3 key must always call up the modal Screen for setting the plant parameters from any page

Proceed as follows:

1. Create a new Shortcut resource from the 'Project Explorer' window and call it 'Main'. This Shortcut resource will always be activated whenever the 'Main' Screen is loaded (by setting the 'Main' Screen as the project's startup Screen, the 'Main' Shortcut will consequently become the project's default Shortcut). The project will not contain other Shortcuts, therefore the 'Main' Shortcut will remain active independently of the Screen currently active.
2. Edit the new Shortcut by adding the commands for:
  - calling up the 'Silos' Screen by pressing the F1 key

- calling up the 'Working Area' Screen by pressing the F2 key
- calling up the 'Main' Screen by pressing the ESC key
- calling up the modal Screen for setting the plant parameters (created beforehand) by pressing the F3 key

# 13. String Table

---

***The Strings Resource contains all the texts which have to be handled in dynamic mode in the project.***

The Movicon strings allow the project to make use of any kind of text in a simple manner. By grouping together a number of texts under a single identifier the resource can handle **Multi-language** functions.

The text strings are needed for handling alarms and messages, for displaying dynamic texts in the screens, and for handling customized popup type windows. When there is the possibility of representing a text dynamically, this text must almost always be contained in the Strings Resource. All the texts typed directly into the drawings or the Movicon objects (i.e. not inserted in the Strings resource) are represented statically in the resource that represents them.

Examples:

1. The text title of an object or symbol can represent a text string which is considered static when typed in directly using the properties window, or can be specified using the Identifier of a text from the String Table. In this case the text will change in accord with the Column (Language) selected.
2. An Alarm or Message, a dynamic text or a pop-up text can be referred to identifiers of strings from the Strings Table.




The String Table can contain a virtually unlimited number of Strings divided up according to columns, whose text can be entered directly in the table fields, or copied or imported from other text applications. String will be saved into a XML file, each one for any language inserted.

Management of text strings as a Movicon Resource makes for an extremely flexible project capable of handling the widest range of applications. One of the main functions of the Strings resource is to contain Alarm and Message texts. When you set an alarm, you will be asked to select a text string from the associated strings resource. Text strings can also be used for displaying information inside pop-up windows. If a pop-up window is associated, it will be formatted according to the length of the text within the workspace.

## Inserting Strings

---

Inserting text strings in the Movicon String Table is done through an appropriated window which is accessed with the **'Edit String Table'** command from the Project Explorer's **"Commands"** window.

The window can then be closed by using the Windows standard techniques such as the  box from the control panel or the CTRL+F4 combo keys.

You can insert, edit or delete strings from the string table by using the purpose-made commands.

A new string is inserted by pressing the right mouse button on the area of the table and then selecting the **'New String'** command. At this point a new String ID will be added with a progressive name. The editing or writing of string texts or string IDs is done by clicking the mouse directly in the box of the table which you wish to edit. The "F2" key can be used for enabling the modifications of the field selected. The TAB key is used for scrolling the different cloumns in edit mode and when reaching the last column passes to the beginning of the next line. This process can be reversed by pressing the SHIFT key.

In addition to this, by clicking with the right mouse key on the string, a menu will appear with the **"Edit Multiline String..."** item. When selecting this item a window will open where you can type in the text using more than one line. Movicon will then insert the right start and end line characters:



The edit window will open on more lines if you click on the string field while keeping the "ALT2 key pressed down at the same time the pressed.

The "\n" character can be inserted at the point where you wish the string to start from the beginning without having to open the "Edit Multiline String" window.



The String Table fully supports the Windows copy & paste command standards using the Windows clipboard. Therefore you can copy the strings from one project to another or paste them in another editor such as Word or Excel. You can also do the reverse in the same way by copying the strings from an editor such as Excel into the Movicon String Table.

The Copy/Paste functions of strings from Movicon to other text editors is carried out with the insertion of separation characters between the various columns (String ID, Language1, Language2, etc.). Movicon inserts the 'TAB' character for default but a different character can be specified (i.e. the "," character) by setting the **"StringSep"** register key.

The String Tables are saved on files in XML format inside the Project folder. These files can then be accessed through ordinary text editors for any eventual editing. Movicon will create a file for each language inserted.

## Special characters

---

The '&' character works in a certain way, and is not always acknowledged as a simple character by Movicon. When it has to be used in a string, or inserted in the title of an object, or whenever you wish to make it appear, it may not be enough to write it once only like all the other characters: Movicon acknowledges and interprets it as a special Shortcut command, whereas Windows uses the '&' symbol placed before a letter to use it as an Shortcut. Therefore in order to write a string with the '&' character you will need to type it twice consecutively. For instance, in order to make the "Start & Go" text appear you will need to type the "Start && Go" string:

String: Start && Go  
Text displayed: Start & Go

## Inserting Missing Strings

---

Movicon provides you with a command to find all the used and missing strings in the different project resources, such as IDs in the String Table. This command, provided for all the project's resources, when executed automatically opens the string editor and adds the missing strings. As normal, only the String IDs will be added leaving the programmer to insert the texts for the different set languages. This allows you to add string IDs to object or resource properties without having to predefine them and then add them to the string table automatically using this command.

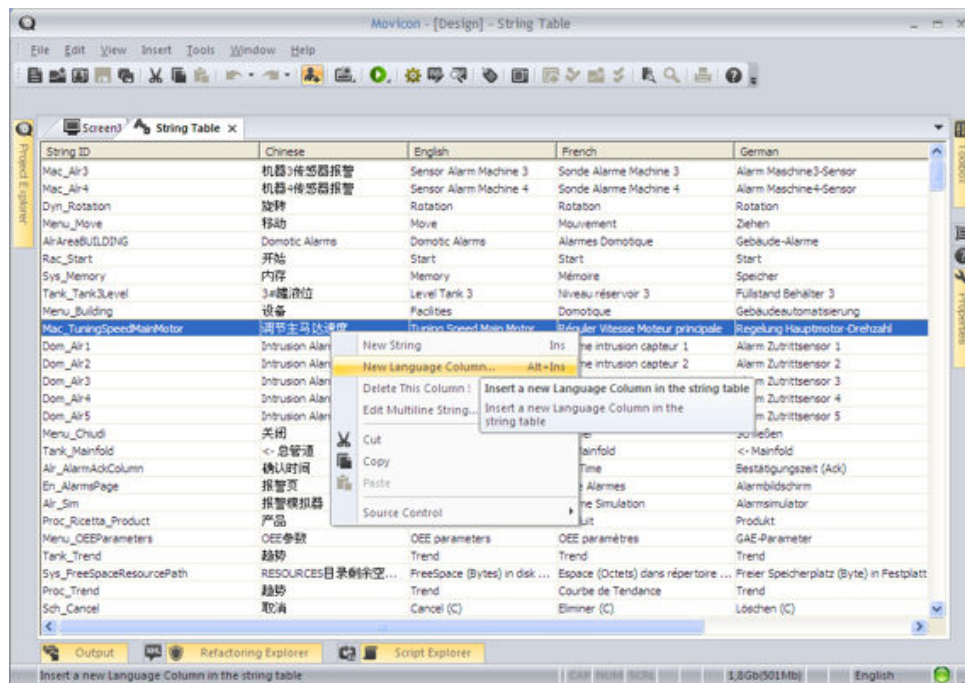
This command, **"Check Missing Strings"**, is available as an item from the 'Edit' and 'View' menus and can be found in icon form on the tool bar.



This command only has effect in those resources that are currently selected in the Project Explorer window or screen (and the objects it contains) currently opened in the workspace. This command supports multi-selections, therefore more than one resource can be selected at the same time (even consisting of a mixture of different resources such as alarms and screens) before executing this command.

## String Selection Window

Each time a text can be inserted into a project resource or object's property you can also select a string from the String table. In order to do this, just click on the (...) browse button to the right of the property field to open a modal window showing the String Table. This window can be used for selecting, modifying or cancelling or inserting a new string or for cancelling or adding a new column.



To modify a string you will need to enter into edit mode in the field of interest with the usual mouse click, F2 key or TAB key. To add a new string or column right mouse click on the table to open a menu.

You cannot use this window to access the columns' properties.



**Column properties cannot be accessed from this string selection window.**

## Changing Languages

The String Table is the container of all the texts in the project which are used to fit any animation requirements. Texts which are typed in directly as titles for the components or drawings, therefore not contained in the String Table, will be presented as permanent fixtures.

The String Table is a container that can be subdivided into columns, if desired, where each one represents a language. When columns are not inserted the resource will be formed by one identifier only and the relative string.

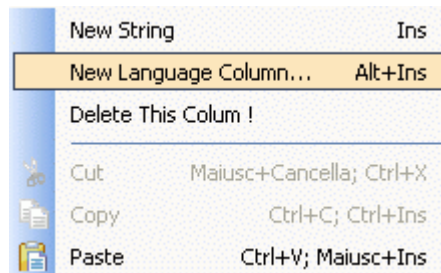
As the columns grow in number, the identifier will refer to different strings according to the column selected as active language.

The following table shows how the columns are divided, where an ID refers to columns with strings in different languages. The enabled column (active selection in the combo.box) will be the one to show the text in the desired language.

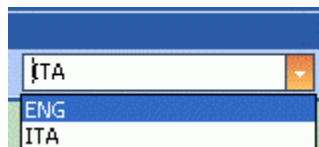


String ID	Chinese	English	French	German
0_menu1	文件	File	Fichier	Datei
0_menu1_2	停止	Stop	Stop	Stop
0_menu2	窗口	Screens	Ecran	Prozessbilder
0_menu3	报警	Alarms	Alarmes	Alarme

To insert a new column press the right mouse button in the area of the table, then select the **'New Language Column...'**. An input-box for inserting the column's name will appear. To change the name of a column (and therefore the language) right click on it's name. To cancel a column right click on one of the column's lines and select the **'Delete this Column!'**.

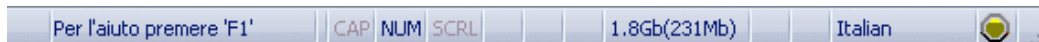


The language to be activated is selected directly from those available in the purpose-built combo-box:

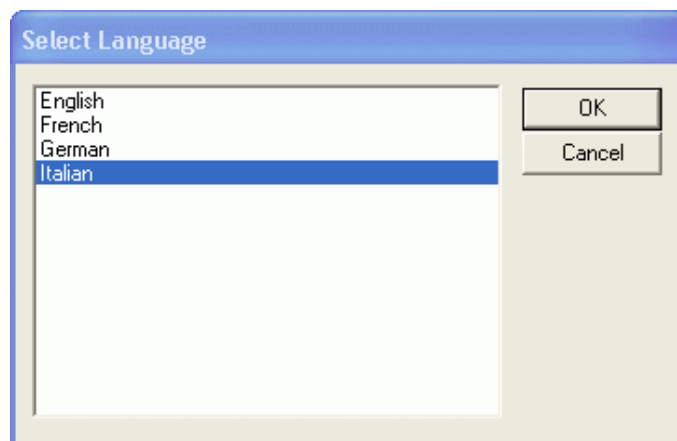


## Project Language Activation

The Movicon Status Bar, whether in design or Runtime mode, shows on the right hand side which language is currently active.



By double-clicking the mouse on the active language field a dialog window will open showing the list of the project's set languages. Here you can select another language and confirm it with OK to activate it.



Selecting a column in Runtime mode, to activate the language desired, can also be done by using the change language' command button from the Movicon "Commands List".

## String formatting through String IDs

---

The String ID management also allows you to link together more than one string ID within a text field. When text is inserted into a field subjected to multilingual, the string ID is searched for in the String Table and replaced with the one relating to the language active. However, if you insert **@(ID)** in the text field, a string with the same ID specified in the brackets will be search for. This will enable you to link together more than one ID string in the same text field.

Example: the following ID strings have been defined in the String Table:

String ID	English	Italian
Alarm	Alarm Active:	Allarme Attivo:
MaxLevel	Tank Max Level!	Livello Max Serbatoio!
MotorOverload	Motor Overload!	Scatto Termico Motore!

By inserting the following syntax in the "Title" property of a rectangle:

@(Alarm) @(MotorOverload)

the following text will be displayed:

Alarm Active: Motor Overload!

Naturally a "@(Alarm) @(MotorOverload)" String ID should not exist in the String Table, otherwise the string associated to that ID will be retrieved redirectly instead of the text format desired.

## Change System Language

---

Movicon has certain dialog windows which are used for customizing texts.

To guarantee the complete support of the multilingual function, Movicon has provided you with a function purposely built for replacing window system texts with customized texts contained in the string resource. In order for this to work you need to insert the customized test, which have been preset with an "key" ID code as described in the table below, into the string resource. When Movicon comes across these special identifiers in the project, it will replace the system's text with the text contained in the string resource according to the language which has been activated.

## Virtual Keypad for Touch screens

---

Special ID	Description
_CANCEL_PADTEXT_	Text for CANCEL key
_DELETE_PADTEXT_	Text for DELETE key
_OK_PADTEXT_	Text for ENTER key
_TITLE_PADTEXT_	Text for the title of the Virtual Keypad
_VALUE_PADTEXT_	Text to replace the "Value" string: on alphanumeric pads.
_LOW_PADTEXT_	Text to replace the "Low limit" string: on numeric pads
_HIGH_PADTEXT_	Text to replace the "High limit" string: on numeric pads.
_CAPSLOCK_PADTEXT_	Text for CAPS LOCK key

## Password Request Window

---

Special ID	Description
_OK_GETPTEXT_	Text desired for OK command
_CANCEL_GETPTEXT_	Text desired for Cancel command
_USER_GETPTEXT_	Text desired for user name request
_PASS_GETPTEXT_	Text desired for password request
_TITLE_GETPTEXT_	Text desired for the title of the Log On window
_LEVEL_GETPTEXT_	Text desired for the password level request <b>Warning</b> , to display the password level request you need to put the %u character after the text specified, ie: Password Level = %u

## Trend Legend

---

Special ID	Description
_TLEGEND_DESC_	"Description" Text
_TLEGEND_VALUE_	"Value" Text
_TLEGEND_MINVALUE_	"Min." Text
_TLEGEND_MAXVALUE_	"Max." Text

_TLEGEND_AVERAGE_	"Average" Text
-------------------	----------------



The change language function also works with the Trend pens by inserting a string ID with the same name as the pen's. In Runtime the Trend will display the text contained in the string ID instead of the name of the pen.

### **Dialog Box for Filter Command in the Viewer Windows (His Log, Trace DB, DataLogger/Recipes)**

ID Speciale	Descrizione
_OK_DBFILTER_	Text desired for OK command
_CANCEL_DBFILTER_	Text desired for Cancel command
_USER_DBFILTER_	Text to replace the "User:" string
_TITLE_DBFILTER_	Text for the title of the Dialog Box
_SORTBY_DBFILTER_	Text to replace the "Sort by:" string
_EVENTTYPE_DBFILTER_	Text to replace the "Event Type:" string
_FROMDATE_DBFILTER_	Text to replace the "From Date:" string
_TODATE_DBFILTER_	Text to replace the "To Date:" string
_SEVCOND_DBFILTER_	Text to replace the "Severity Condition:" string
_SEVERITY_DBFILTER_	Text to replace the "Severity:" string

### **Dialog Box for insert Trace Comment for a variable**

ID Speciale	Descrizione
_OK_TRACECOMMENT_	Text desired for OK command
_CANCEL_TRACECOMMENT_	Text desired for Cancel command
_NAME_TRACECOMMENT_	Text to replace the "Name:" string
_VALUE_TRACECOMMENT_	Text to replace the "Current Value:" string
_CHANGING_TRACECOMMENT_	Text to replace the "Changing Value:" string
_OBJECT_TRACECOMMENT_	Text to replace the "Changing Object:" string
_COMMENT_TRACECOMMENT_	Text to replace the "Comment:" string
_TITLE_TRACECOMMENT_	Text for the title of the Dialog Box

# Language Column Properties

---

The properties which can be associated to each language (String Table columns) allow you to define certain settings which will be used in the runtime phase when the language is activated. These settings are then used by the Alarm Dispatcher to send messages.

To display the column properties just select a string in the desired column and the Properties Window will get updated with the column's properties.

## General Language Column Properties

---

The General properties allow you to define settings to do with the Style References for objects and TTS setups for the speech functions.

To change the General properties, click on a string in the column desired and use the Movicon **"Properties Window"**.

### Language

Read only property which shows the name of the selected column.

### Style Source Container

This is used for selecting the reference screen for the style management. For further information on how these properties work please see "Style Sources in Symbols".

### TTS Engine

This property is used for selecting the item to be used for the text speech. The items which can be selected are the same ones provided in the operating system which can be verified in its "Control Panel - Speech". The moment the language is activated the voice associated to it is automatically loaded to enounce texts.

This setting is used also for sending messages through the alarm Dispatcher based on the language associated to the message recipient user.

### TTS Volume

This property is used for setting the speech volume percentage. This percentage will be applied to the volume level set in the PC. The end volume level will depend on different factors, speaker volumes, volume set in the Control Panel, TSS Volume property settings.

This setting will also be used for sending messages with Alarm Dispatcher based on the language associated to the message recipient user.

### TTS Rate

This property is used for setting the speech speed. The 0 to 10 value will make the speech faster, while the 0 to -10 value will slow it down.

This setting will also be used for sending messages with Alarm Dispatcher based on the language associated to the message recipient user.

# 14. Variables Scaling

*The Variable Scaling resource tool is needed when carrying out automatic scaling of data, which entails converting numerical values into engineering units.*

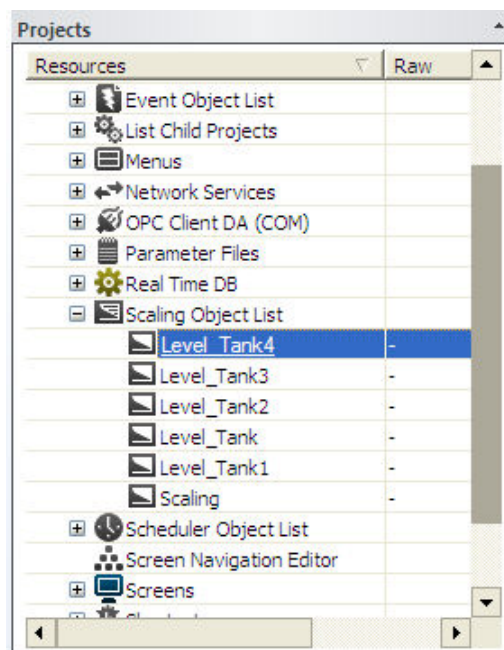
Variables are very often acquired in a project from the plant which correspond exactly to the digitally converted raw values of measured physical quantities.

This value must be converted or 'scaled' in 'engineering units', which is a value corresponding to the measured physical quantity. This involves carrying out calculations for transforming, nearly always linear, from a minimum and a maximum of the digital value to a minimum and maximum scaled value.

Movicon allows these conversions to be done directly through the **"Variable Engineering Data Properties"**, so that the variable directly gets the scaled value arriving from the field.

It may also be necessary at times to carry out the scaling between two variables within the project, or to do NON linear conversions which entails the use of the 'Variable Scaling' resource through which the two variables can be specified, the one to be scaled and the scaled one, to get NON linear conversions.

This resource is available in the Movicon 'Project Explorer':



A virtually unlimited number of objects can be inserted into the Variable Scaling resource, each one will be scaled between two variables.

To add, copy or cancel Scaling objects use the standard techniques described in the paragraph on **"The Resources"**.



The main properties of Scaling objects can also be edited in the Runtime phase through the appropriate Basic Script functions.

## Variable Scaling Properties

---

The Variable Scaling Properties are used to determine the input variables to be scaled and the scaled output ones, by associating the conversion or linear or non linear scaling factors.

To edit the properties of Variable Scaling objects, select the object with the mouse and use the Movicon **'Properties Window'**.



Scaling is **bidirectional**: in normal operations, the variable to be scaled is scaled on the output variable, and scaled on the input variable in inverse operations. Movicon executes inverse operations only when the scaled output variable undergoes a change.

### Variable Scaling General Properties

---

The General properties are used to determine the input variables to be scaled and the scaled output ones, by associating the conversion linear or non linear scaling factors.

To edit the property of Variable Scaling objects, select the object with the mouse and use the Movicon **'Properties Window'**.

#### **Name**

The name to be assigned to the scaling object is entered in this edit box. Each object must have a unique name within the list so that they can be referred to through the Basic Script functions in order to be used during Runtime.

#### **Enable**

This selection box has to be enabled to activate the scaling object in question otherwise the conversion operations will not be executed.

#### **Raw Variable**

The name of the variable containing the value to be scaled is entered in this edit box (or selected with the '...' browse button on the right).

#### **Scale to Variable**

The name of the variable containing the scaled value, being the calculated result of the conversion based on conversion factors set in the property described below, is entered in this edit box (or selected with the '...' browse button on the right).

#### **Dead Band**

This edit box is used to establish a dead band in the conversion factors. The dead band establishes the value to which the scaled variable is to be set when the non raw variable value goes over the set conversion limit. The default value set by Movicon is '-1'.

#### **Raw Min.**

The minimum value of the raw variable is specified in this box, being the input value. The minimum and maximum linear scaled output value will be calculated based on the minimum and maximum input value.

#### **Raw Max.**

The maximum value of the Raw variable is specified in this edit box, being the input value. The minimum and maximum linear scaled output value will be calculated based on the minimum and maximum input value.

This value will represent the first non linear segment when non linear scaling is being used. The next segments are entered through the **"Non Linear Scaling"** window.

#### **Scale to Min.**

The minimum value of the scaled variable is specified in this box, being the output value corresponding to the real physical quantity.

#### **Scale to Max.**

The maximum value of the scaled variable is specified in this box, being the output value corresponding to the real physical quantity.

This value will represent the first non linear segment when non linear scaling is being used. The next segments are entered through the **"Non Linear Scaling"** window.

**Non Linear Scaling**

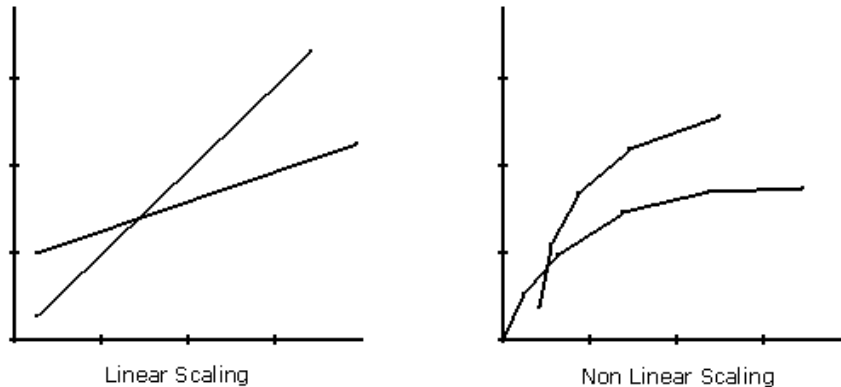
Non Linear Scaling can also be used if required. The segment settings are accessed through this button where they can be established between the min. and max. values. This will create a theoretic line broken up into a number of segments.

For further information refer to the paragraph **"Non Linear Scaling"**.



# Non Linear Scaling

Movicon consents the scaling of non linear values, which are those whose behaviour does not follow an exact straight line between the minimum and maximum values.



To insert non linear conversions, access the values settings by using the **"Non Linear Scaling"** button from the **"Variable Scaling General Properties"**.

In this case, the maximum values set previously assume the first segment of the non-linear curve. The proceeding segments are set through the window as illustrated below:

The 'Non Linear Scaling' dialog box contains the following fields and controls:

- Raw Max. Value: 3000
- Scaled Max. Value: 350
- Buttons: Add, Edit, Delete
- Table with 2 columns: Raw Max. Value, Scaled Max. Value
- Table data:

Raw Max. Value	Scaled Max. Value
2000.00	200.00
3000.00	350.00

At the bottom of the dialog are buttons for OK, Cancel, and a help icon (?)

After establishing the first segment from the maximum values set in the **"Value"** property in the **"Variable Scaling General Properties"**, the other segments can be added by entering the following maximum values and clicking the **'Add'** button. This will establish an equivalence between the two segments.

The **'Edit'** button is used for changing the previous entered. The **'Delete'** button is used for deleting a previous entered value.



The values comprised in one single segment will be scaled in linear mode, therefore the more segments inserted the more the line will become curved.

# 15. Command Scheduler

---

*The Command Scheduler resource tool is used for configuring timed command executions.*

The scheduler is a very handy tool to have in projects when commands need to be activated on preset times, especially on plants where commands must be activated at a certain time and for a certain period.

Time is also noticeably reduced when creating projects where commands have to be repeatedly activated on the plant at preset time intervals.



The scheduler's job is to execute commands or list of commands based on preset times respecting the PC system's clock.

This resource is available in the Movicon 'Project Explorer' window:



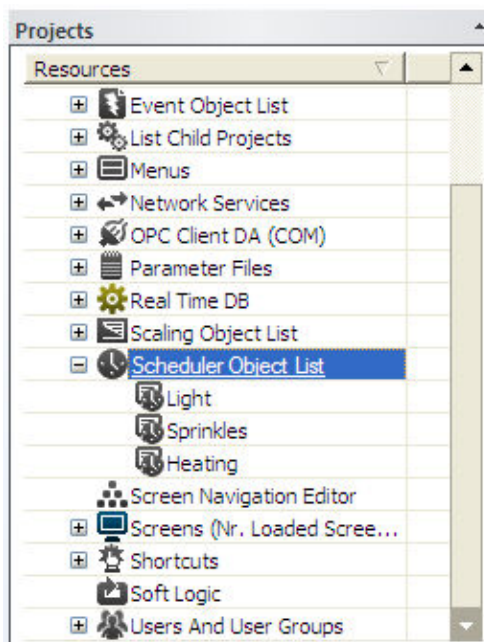
This resource can contain a virtually unlimited number of commands which can be executed according to preset times or preset weekly schedules, which are edited through the **'Properties Window'**.

The Command Scheduler resource offers great operating advantages in any application, especially in those sectors (ie. Building Automation) which normally require commands to be automatically executed on preset times.

A few of the most common uses would be for example the commands for turning on/off of lights, central heating, etc.

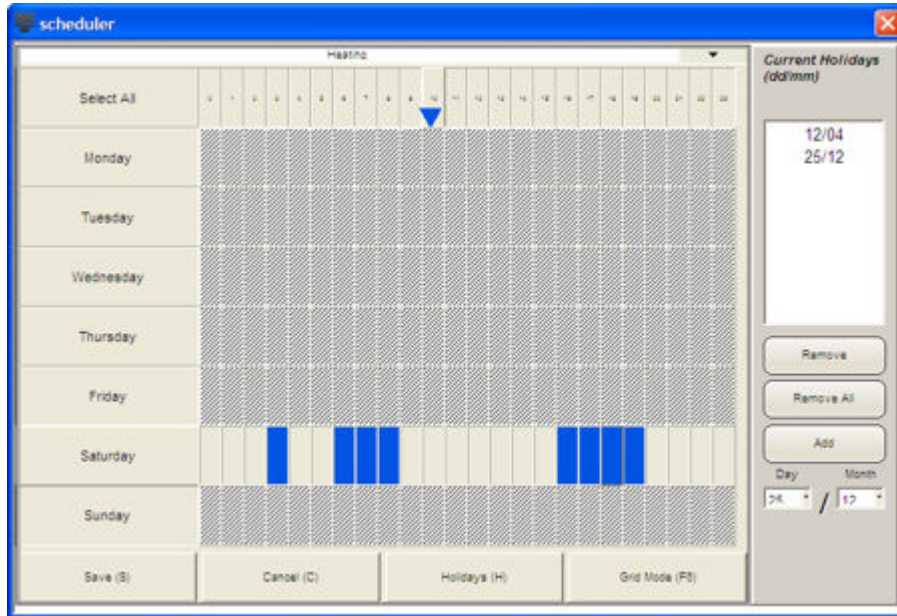
These simple operations can therefore be realized through this purposely built resource which has the job of slimming down and speeding up the procedures for setting these type of commands.

To add, copy or cancel a Scheduler object use the standard techniques described in the paragraph on **"The Resources"**.



## Adding new Schedulers at runtime

The Hour Selector object allows, during runtime, to add new Schedulers and to remove them, if required, using the dedicated command buttons ("Add Scheduler Button", "Remove Scheduler Button"). Please note that the Schedulers created at runtime can manage only the "Commands On": if used to manage the set of a variable, you should also add the time intervals when the variable should be reset.



## Schedulers retentivity files

Schedulers set as "Daily Plan" create during runtime a retentivity file named "<ProjectName>\_<SchedulerName>.sst" in the "DATA" project folder. The retentivity file purpose is avoiding to repeat the ON command, if this command has already been executed once. If, while the project is starting, the ON command is modified by a time interval of the Scheduler, which sets the value or the name of the variable associated to command, the ON command will be executed anyway.

## Holiday Planning

The Scheduler objects can be set to manage holidays as well. When a Scheduler is enabled through the relating property to do this, it will be able to memorize, internally, the dates set as holidays. Therefore two types of planning, one for normal planning and one for holiday planning, can be set by means of using the "Scheduler Window" graphic object. The holidays are controlled only when the Scheduler is a 'Date' type or 'Daily plan' type and only the day and month of the holiday date are controlled (if they correspond with the current day and month), while the rest of the date field is not valuated.

The holidays are only managed when the Scheduler is 'Date' type or 'Daily plan' type in the two following modes:

- **Daily Plan:** By means of using the "Scheduler Window" object you can set two different daily plans: normal and holidays. The Scheduler object's "Holiday Button" to switch from one daily plan to the other. The hours in both of the two plans are composed of a 7 day by 24 hour grid. The holiday daily plan settings are in periods of 15 minutes. The periods selected in the holiday daily plan are represented in the default grey colour; while the normal ones are in blue. At this point the Scheduler will use the appropriate daily plan according to whether the day has been set as a holiday or not

- **Date:** The scheduler will execute the preset command, not only for the specified date but also for the date set as a holiday

## Setting Holidays

In order to set holidays you need to use some of the VBA methods from the "SchedulerCmdTarget" interface. The methods available for managing holidays are as follows:

AddHoliday: allows a holiday to be added to the scheduler object  
 RemoveHoliday: allows a holiday to be deleted from the scheduler object  
 HasHolidays: lets you know in write whether the scheduler object has been set for managing holidays  
 IsHoliday: allows you to verify whether a date has been set as a holiday in the scheduler object  
 GetHolidaysString: returns a string divided by a pre-chosen separator character (parameter from function) with the list of all the holiday dates set in the scheduler object  
 HolidaysPlan: allows you to read the current weekly daily plan from a "Daily plan" scheduler  
 SaveRetentive: allows you to save the daily plans (normal or holiday) and the days of the holidays on external files to be retained even after an application re-startup

You can however set a list of holiday dates also through editing XML files. These files must be saved in the project folder, with the same name of the project and with the ".defhol" extension. All the "fixed Date" type schedulers, which do not have holiday lists, will use those defined with the XML file with the ".defhol" extension as their holidays. When a ".defhol" file cannot be found in the project folder, it will be searched for in the Movicon installation with the "Holidays.defhol" fixed name.



**the XML file with the holidays list is read only at the project startup and therefore modification to the file will not be managed until the project is started up again.**

The Holiday XML file structure must be in the same way as:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<DefaultHolidays>
  <HolidaysDates
    n0="2000-01-01T00:00:00Z"
    n1="2000-01-06T00:00:00Z"
    n2="2000-04-25T00:00:00Z"
    n3="2000-05-01T00:00:00Z"
    n4="2000-06-02T00:00:00Z"
    n5="2000-08-15T00:00:00Z"
    n6="2000-11-01T00:00:00Z"
    n7="2000-12-08T00:00:00Z"
    n8="2000-12-25T00:00:00Z"
    n9="2000-12-26T00:00:00Z"
  />
</DefaultHolidays>
```

This example shows the Italian national holidays.

# Command Scheduler Proprieties

---

The Command Scheduler Properties allow you to select the execution times and commands to be activated.

To edit the Scheduler object properties, select the object with the mouse and use the Movicon **'Properties Window'**.

## Command Scheduler General Properties

---

The General properties allow you to select the execution times and commands to be activated.

To edit the General properties, select the object with the mouse and use the Movicon **'Properties Window'**.

### Name

The name of the scheduler object is entered in this edit box. Each object must have its own name within the list. This is very essential when each object must be referred to singularly through the Basic Script functions so it can be used during Runtime.

### Enable

When enabling this selection box the Scheduler object in question will be activated. If this box is left disabled any eventual commands will not be executed.

### Support Holidays

You can activate the "Holidays Planning" management for the Scheduler object in question by using this selection box. For further information about "Holidays Planning" please refer to section with the same title.

### Treat Holidays as Sunday

This option box, if enabled, consents the normal Sunday hour plan be used for managing the days set as Holidays, being the ones set in the scheduler's holiday list. This option does not oblige the user to set a seven day hour plan for the holidays but helps in simplifying work when all the holidays of one scheduler must be managed in the same way, independently from the day of the week.

### Enable Variable

This edit box allows you set a variable for enabling the Scheduler. The Scheduler object must, however, be enabled in development mode with the appropriate **"Enable"** property, otherwise it will always be disabled in runtime.

### Type

The Scheduler object's purpose is to execute one or more Movicon commands on preset time periods.

The "Type" selection box permits you to select in which moment the command must be executed. Other fields may have to be set according to the item selected in order to complete this type of programming. For instance, when selecting the **'Every Minute'** or **'Every Hour'** item it is not necessary to specify anything else, as it is quite explicit that the onset of each new preset period the associated event will be automatically executed.

However when selecting a day or a month, it is then necessary to indicate, inside that period, at what precise time the command must be executed. The time must therefore be set in the **'Time'** property.

Amongst the scheduled time settings to be assigned you will find the **'Date'** setting. This selection allows the activation of a virtual calendar, through which you can select a date up to the year 9999. The moment the selected date is verified (after midnight of the previous day) the event will be executed. The date in question must be set in the **'Time'** property.

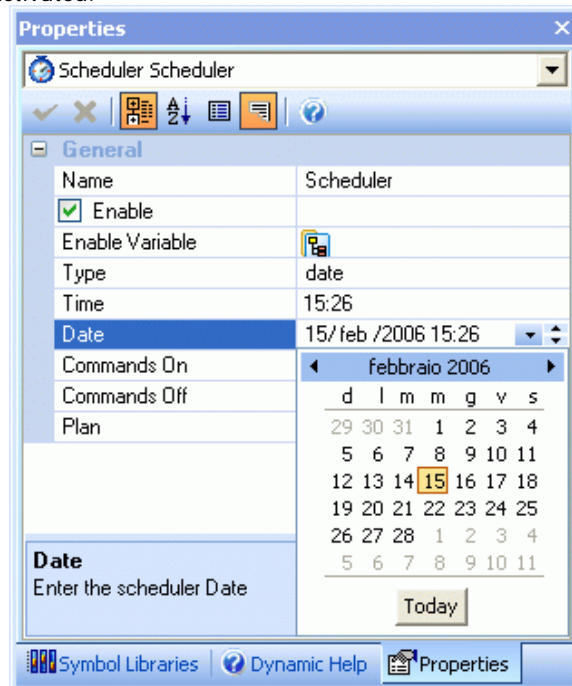
The **'Daily Plan'** setting activates a weekly plan. When selecting this setting it will be necessary to access the weekly plan scheduler setting through the **'Plan'** property.

### Time

The command activation time is entered in this edit box.

### Date

This field is enabled only when the "date" item has been selected in the "Type" property. In this case, when clicking on this box a window will open for selecting the day when the command is to be activated.



The presence of the "Date" field distinguished by the "Time" field permits two things:

- To set the command execution time in "Date" type schedulers
- To make sure that the command is executed on a certain day in each year. Movicon only controls the month and day and not the year set

### Commands On

This button opens a Movicon 'Command List' through which you can set a list of one or more commands to be executed by the scheduler when the time goes On.

For further information on the commands available please consult the paragraph on "Command List".

### Commands Off

This button opens a Movicon 'Command List' through which you can set a list of one or more commands to be executed by the scheduler when the time goes Off.

For further information on the commands available please consult the paragraph on "Command List".

### Plan

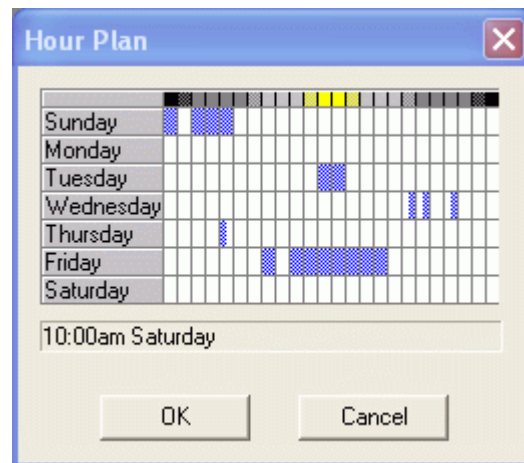
The daily plan window allows you to configure the time ranges in which the command associated to the Scheduler must be executed.

The table provides seven lines corresponding to the days of the week, and 24 columns corresponding to the hours in a day.

Left click on the squares to select the day and hour required. Click on the square again to deselect.

Right click to select the precise time of each hour, each click equals 15 minute resolutions.

Click on the day to select the whole 24 hours for that day.

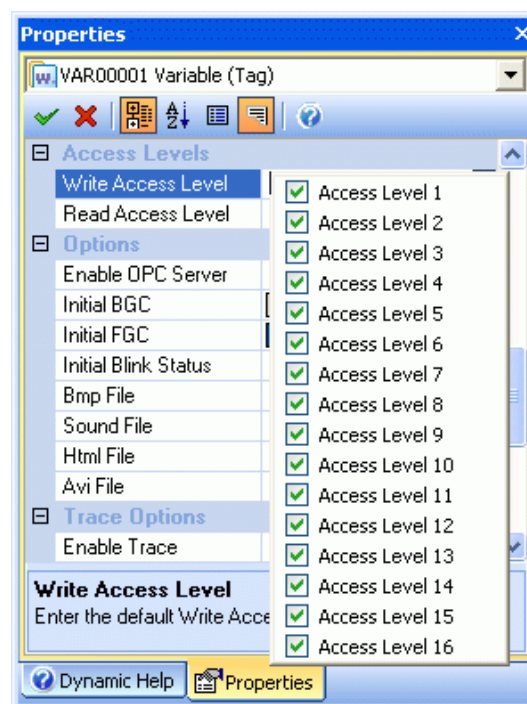


## Scheduler Access Level Properties

By using this Access Level properties you can enable which are the default write access levels for the selected Scheduler.

To modify the Access Levels of a Scheduler, select the object with the mouse and use the Movicon **"Property Windows"**.

There are 16 access levels available and they can be selected individually one by one:



### Write Access Level

By using this property you can define the Access Level mask for modifying the Scheduler through the "Scheduler Window". By using the "Scheduler Window" you can in fact select a Scheduler and display or modifying its plan. However, if the password management has been enabled and the "Always Visible" property disabled, the modifications to the Scheduler's plan will only be applied if the active user has the right Access Level requested by the Scheduler's "Write Access Level" mask. Otherwise the "Scheduler Window" will remain disabled as far as the plan editing commands are concerned.

For further information on 'Access Levels' please refer to **"User Levels and Access Levels"**.

#### **Read Access Level**

By using this property you can define the Access Level mask for selecting the Scheduler the "Scheduler Window". By using the "Scheduler Window" you can select a Scheduler through an appropriate list-box and display or modify its plan. However, if the password management is enabled and the "Always Visible" property disabled, the Scheduler can only be selected if the active user has the right Access Level requested by the Scheduler's "Read Access Level" mask. Otherwise the Scheduler will not be listed in the selection listbox and therefore will not be available for selecting.

For further information on 'Access Levels', please refer to the paragraph on **"User Levels and Access Levels"**.

#### **Always Visible**

When enabled, this property allows the Scheduler to be selected and modified through the "Scheduler Window" independently from the active User. In cases where the Password management has been disabled, it will be necessary to enable this property in order to view the Scheduler in the "Scheduler Window" (therefore selectable and modifiable).

For further information on 'Access Levels', please refer to the paragraph on **"User Levels and Access Levels"**.

## **Command Scheduler Example**

---

Let's suppose you need to configure your project so that every evening at 21.00 all the plant's lighting is automatically turned on (subject to the 'ILLU1' variable) managed by the supervision, and automatically turned off in the morning at 6.30.

First of all you need to proceed with editing the scheduler objects by selecting the 'Scheduler object list' resource from the 'Project Explorer' window to add the scheduler command. Go ahead and edit the Movicon scheduler objects as follows:

1. Select the 'Scheduler object list' from the 'Project Explorer' window
2. Use the right mouse button to insert a new object in the Scheduler resource using 'Add New Scheduler Object' command
3. From the object's General Properties window set the 'Type' property with 'Every Day' and set the time at '21.00' in the 'Time' box. Then enter 'Turn On' as name
4. By using the 'Command' property open the 'Command List' window and select the 'Variable command'. Enter the variable required, in our example we will use the 'ILLU1' variable, and enter 'Set' as Action and enter '1' in the 'value' property
5. Confirm with OK to enter the first scheduler object called 'Turn On'. Now continue with entering the second one which will be called 'Turn Off'
6. Right mouse click on the 'Scheduler Object List' resource and insert a new object using the 'Add New Scheduler' command
7. From the object's General 'Properties Window' set the 'Type' property to 'Every Day' and set '06.30' in the 'Time' box. Then assign the object with the name 'Turn Off'
8. Click the 'Command' property to open the 'Command List' window and select 'Variable Command'. Insert the variable required, in our example we will use the 'ILLU1' variable, and set the 'Set' value to '0'
9. Confirm with OK to enter the second and last Scheduler object

At this point the plant's light management has been completed.

When running the project the scheduler will execute the corresponding command automatically when the set time has been reached and verified.

The time set in the scheduler refers to the PC system's clock.





# 16. Commands on Event

---

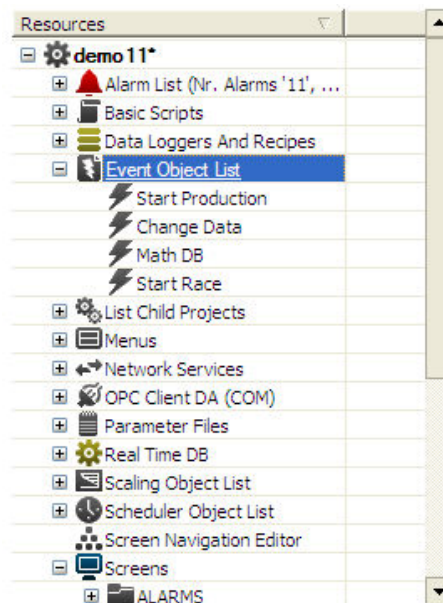
*The Event Objects resource is the tool which allows commands to be executed according to Movicon variable changes.*

The Commands on Event are very handy when values of one or more variables must be kept monitored in the project on which command executions are to be based.



The Event Objects therefore have the job of executing the command or list of commands according to the value obtained by a certain variable.

This resource is available in the Movicon 'Project Explorer' window.



This resource can contain a virtually unlimited number of Event Objects, each one of which is associated to a different variable, which can be edited by means of the **'Properties Window'**.

The Event Object resource offers great operating advantages in any application where variable values need to be controlled to execute one or more commands upon value changes. This enables the programmer to avoid writing codes need for the controls of the variables in question. To add, copy or cancel an Event Object use the standard techniques described in the paragraph titled **"The Resource"**.

## Event Object Properties

---

The Event Object properties are used for selecting the variables to be monitored and commands to be activated.

To edit the Event Objects' properties, select the object the mouse and use the Movicon **'Properties Window'**.

## Event Objects General Properties

---

The General properties are used for selecting the variables to be monitored and the commands to be activated.

To edit the General properties, select the object with the mouse and use the Movicon **'Properties Window'**.

### Name

This edit box is used for assigning a name to the Event Object. Each object must be given a unique name in the list. This is essential for referring to each object through the Basic Script functions so that they can be used during Runtime.

### Enable

Checking this selection box will activate the Event Object in question.

### Variable

The name of the variable which is to be monitored is entered in this edit box (or selected with the '...' browse button on the right). When the variable's value changes, if requested by the 'Condition' property settings, the associated **'Command'** will be executed.

### Enable Variable

The name of the variable which determines the dynamic enabling of the Event Object is entered in this edit box (or selected with the '...' browse button on the right). When this variable's value is equal to 'zero' during Runtime, the Event Object's **'Command'** will not be executed. When its value is different from 'zero' the Event Object's 'Command' will be executed according to the set conditions.



This functionality is managed by Movicon only when the **'Enable'** property has been activated. If otherwise the Event Object will not execute any commands whatever the **'Enable Variable'** state if the 'Enable' property has not been activated.

### Condition

This property is used for establishing which type of variable value's change executes the command. The possibilities are:

- **Change:** the command will be executed every time the variables changes value
- **Major:** the command will be executed every time the variable's contents passes from a low or the same value specified in the **'Value'** property to a higher value
- **Minor:** the command will be executed every time the variable's contents passes from a high or the same value specified in the **'Value'** property to a lower value
- **Equal:** the command will be executed every time the variable's contents pass from different value to that specified in the **'Value'** property to one with the same value

### Value

This edit box is used for specifying which threshold value is to be referred to by the **'Condition'** property. This setting is insignificant when the **'Change'** option has been selected in the **'Condition'** property.

### Commands

This button opens the **'Command list'** to setup the list of one or more commands which are to be executed by the Event.

For further information on the available commands please consult the paragraph titled **"Command List"**.

### Run At Server

When the Event is run on command, by means of the Command List, in a Client project, with this option enabled (default), the Event object's commands will only be run in the Server project. If this option is enabled, the commands will only be run in the Client project. This option only has meaning when used in a Client project created with the child project techniques or in Redundancy.

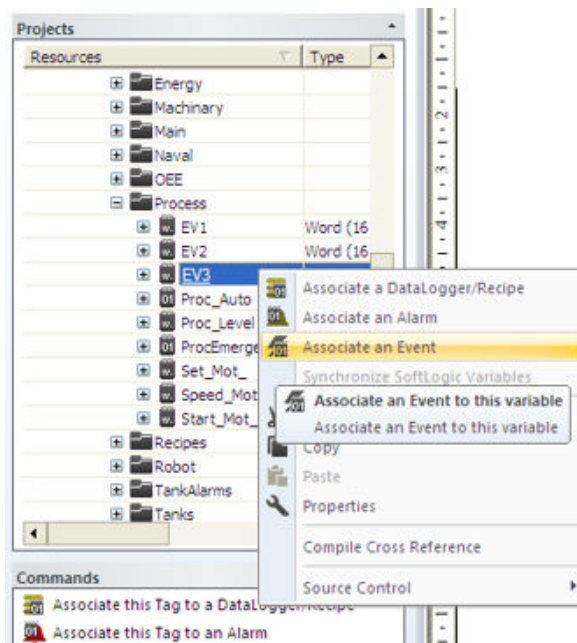
# Events as Templates

Movicon has an extremely useful function when needing to set many repetitive Event objects in the project, which record and sample series of different variables in the same way.

- **A practical example would be:** we have to get our project to manage 50 data recordings with the same sampling time and recording modes. The RealTime Database has declared the 50 variables exchanged with the field and relating to the process information. These variables must generate the same type of event, set in an event object.

In a case such as this, we need a tool which to quicken the process of assigning Tags to the Event object, one which provides you with the possibility to set up the event object **only once** and associate it to the 50 variables, of the example above, all at once. A tool such as this exists in Movicon and is called "**Event Template**".

To get this function you have to set the Event as "Template" type in the Event Object Resource along with the activation modalities and command desired. Then you can select the variables needed from the Variables List from the RealTime Database and use the "Associate an Event" command with the right mouse key and select the event window desired from the window that pops up. The variables (in our example this would be the 50 process variables) will then be linked to the Event Template. Technically Movicon has set 50 different variables linked to the same Event in its project on receiving just one click of the mouse.



The request to associate an Event will open another window containing a list of Events, which should have been inserted beforehand.



**CAUTION:** there shouldn't be any reference variable specified in the properties of the event to be associated but only the activation condition and command list.



# 17. Objects Commands List

---

*Movicon provides a list of commands which can be executed during Runtime from both the resources and controls.*

The Movicon resources and controls predisposed for executing commands are set up through the '**Command**' property found in the '**Properties Window**' of the object selected.

By activating the '**Command**' property the '**Command List**' will open through which the required commands can be inserted, edited or cancelled.



The command objects usually provide the possibility to permit or deny the operator command execution by means of associating the object with a password and access level variable.

The execution of any command can only take place during Project Runtime mode.

## Command List Window

---

The "**Command List**" window lists the commands that are already inserted in the selected resource or control.

To add a new command to the list use the '**New Command...**' button. The "**Command Type**" window will open to select and configure a new command.

To edit the new command, just inserted into the list, select the command with the mouse and click on the '**Edit**' button to open the "**Command Type**" window.

To delete commands from the list simply press the '**Delete**' command.

## Variable Commands

---

This group of commands is used for carry out operations on the Movicon Real Time DB variables.

### Variable

The variable name on which the required command is to be activated is enter in this edit box (or selected with the '...' browse button on the right).

When selecting a child project variable the syntax is:

**ChildProjectName\VariableName**

When selecting a parent project variable from a child project the syntax is:

**..\VariableName**

### Action

The command type or action to be executed on the variable, is selected through this property. The options are:

Action	Description
set	This command sets the variable to the value specified in the ' <b>Value</b> ' property. When the variable is string type, any numeric values will automatically be converted in string. When the ' <b>Value</b> ' field is left empty, the ' <b>set</b> ' command will have no effect on numeric type variables, while an empty

	string will be inserted if the variable is string type.
<b>reset</b>	This commands sets the variable to '0' value (zero) independently of the ' <b>Value</b> ' field's contents. When the variable is string type, an empty string will be inserted. In addition to this, this command also sets the "append decimal mode" to OFF (ON-OFF decimal append values). In this way the next command used for appending a new value will always insert the value before the decimal point. This change makes it easier to use Numeric Pads created with Screens (Numeric and Alphanumeric Screens).
<b>toggle</b>	This command, changes the variable's value upon each execution according to the previous state. The two alternating values are the '0' value and the value set in the ' <b>Value</b> ' property. When the variable is string type, an empty string will be inserted instead of the '0' value.
<b>strobe</b>	This command sets the variable to the value contained in the ' <b>Value</b> ' property, and is kept for the time specified in the ' <b>Strobe Time</b> ' property. When this time runs out the previous variable value will be restored.
<b>increase</b>	This command increases the variable specified in the ' <b>Variable</b> ' property by one value which is specified in the ' <b>Value</b> ' property. When the variable is string type a chain of strings will be executed instead, adding in line to the ' <b>Value</b> ' field's contents. The increasing of the variable will only have effect up to the maximum value, set in the ' <b>Max. Value</b> ' property for numerics and ' <b>Max. Chars</b> ' property for strings. When the ' <b>Value</b> ' field is left empty, the ' <b>increase</b> ' command will have no effect on the variable.
<b>decrease</b>	This command decreases the variable specified in the ' <b>Variable</b> ' property by one value which is specified in the ' <b>Value</b> ' property. The variable decrease will only have effect until the minimum value set in the ' <b>Min. Value</b> ' property has been reached. A null string will be inserted when the variable is string type.
<b>alphanumeric pad</b>	This command opens a alphanumeric pad where values can be entered without using the keyboard. The value will return to the variable after being confirmed with the 'Ok' key as usual. This is fundamentally used with Touch-Screens. If the name of a screen has been specified in the project's "Alphanumeric Screen" execution property this screen will be opened instead of the default Alphanumeric Pad. In this way the Alphanumeric Pad can be customized. For further information please refer to the section on "Numeric and Alphanumeric".
<b>numeric pad</b>	This command opens a numeric pad where characters can be entered without using the keyboard. The value returns back to the variable after being confirmed with the 'OK' key as usual. This is fundamentally used with Touch-Screens. If the name of a screen has been specified in the project's "Numeric Screen" execution property this screen will be opened instead of the default Numeric Pad. In this way the Alphanumeric Pad can be customized. For further information please refer to the section on "Numeric and Alphanumeric".
<b>Append Value</b>	This command appends the value specified in the "Value" field to the selected variable. This command does not execute a adds up the values but adds the specified value to the end of the current value. If, for instance, the variable

	<p>has a "10" value and the "1" value is to be appended the result will not be "11" but "101". Naturally the value to be appended must be a numeric figure if the variable type is numeric, or otherwise a alphanumeric figure if the variable is string type.</p> <p>If a value has not been specified in the "Value" field Movicon will append the value which has been set in the "Title" of the object which executed the command (eg. the button's title).</p>
<b>Remove Value</b>	This command removes the last figure from the selected variable. If, for instance, the variable's value is "101" it will then become "10" when this command is executed.
<b>Swap Plus-Minus</b>	<p>This command allows the variables value to swap over from positive to negative and viceversa.</p> <p>This command has effect only when the variable type is numeric. This command it toggle type, meaning that the variable's current sign is inverted with the other sign each time it is executed.</p>
<b>Append Decimal Mode ON-OFF</b>	<p>This command allows you to specify whether the "Append Value" or "Remove Value" command must be executed on the variable in integer part or fractional part. This command has meaning only when the variable is floating point type.</p> <p>This command is toggle type, meaning that it inverts current position to the other each time it is executed.</p>
<b>Move Value</b>	Allows a value from one variable to be moved to another variable. In this case the source variable is the one specified in the "Variable" parameter, while the target variable is the one specified in the "Move to Variable" parameter. Different variable types can also be used. In this case Movicon will execute the necessary conversions according to each variable type. For instance, you can move the value of a string value to a numeric value and viceversa.
<b>Reset Statistic</b>	This command will reset all the statistic data of the variable set in the "Variable" parameter. This operation can also be executed with VBA.
<b>Move Min.Value</b>	This command copies the minimum value statistic of the variable indicated in the "Variable " parameter to the variable indicated in the "Move to Variable" parameter. Therefore the destination variable will get the minimum value that the source variable has obtained during the project run.
<b>Move Max. Value</b>	This command copies the maximum value statistic of the variable indicated in the "Variable" parameter to the variable indicated in the "Move to Variable". Therefore the destination variable will get the maximum value obtained by the source variable during the project run.
<b>Move Average value</b>	This command copies the average value statistic of the variable indicated in the "Variable" parameter to the variable indicated in the "Move to Variable" parameter. Therefore the destination variable will get the average value obtained by the source variable during the project run.

#### Move to Variable

The name of the destination variable is entered in this edit box (or selected with the browse button on the right) when selecting a **"Move Value"** command. This is the variable on which the value of the source variable specified in the "Variable" property, will be moved to according to command type.



When selecting a child project variable the syntax is:

**ChildProjectName\VariableName**

When selecting a parent project variable from a child project the syntax is:

**..\VariableName**

#### **Value**

The value to be obtained by the variable when the command is executed is entered in this edit box. Its significance changes according to the type of **'Action'** the variable is associated with.

#### **Strobe Time**

This property is only valid when **'Strobe'** has been selected in the **'Action'** property. In this case the time, expressed in milliseconds, in which the variable will remain set with the value entered in the **'Value'** property. When this time runs out the variable will go back to previous value again.

#### **Max. Value**

This property is only valid when the following items have been selected in the **'Action'** property:

- **Increase**
- **Numeric Pad**

The specified value is the maximum limit that the variable can assume during increase operations or Numeric Pad entries.

The name of the variable from the RealTimeDB can also be inserted in this field whose contents will be taken as the value limit rendering the value limit dynamic and can be entered using the numeric pad.

#### **Min. Value**

This property is only valid for numeric type variables and when the following items have been selected in the **'Action'** property:

- **Decrease**
- **Numeric Pad**

The specified value will be the minimum limit which the variable can assume during decrease operations or Numeric Pad entries.

The name of the variable from the RealTime DB can also be inserted in this field whose contents will be taken as the value limit rendering the value limit dynamic and can be entered using the numeric pad.

#### **Max. Chars**

This property is only valid for string type variables and when the following items have been selected in the **'Action'** property:

- **Increase**
- **Alphanumeric Pad**

The value specified will be the maximum number of characters which can be entered in the variable during increase operations or Alphanumeric Pad entries.

## Screen Commands

---

This group of commands are used for carrying out opening operations of the Movicon Screen windows.

### Screen

The name of the Screen required to be opened, printed or closed is entered in this edit box (or selected with the '...' browse button on the right).

When selecting a child project screen the syntax is:

**ChildProjectName\ScreenName**

When selecting a parent project screen from a child project the syntax is:

**..\ScreenName**

### Action

The type of command or action to be executed on the selected Screen is selected through this property. The options are:

Action	Description
<b>Open Normal (screen change)</b>	This command consent a Screen window to be opened in the more classic mode being a window within the Movicon Main frame. Opening the Screen in this way will automatically close the previous Screen (which is loaded in the system's RAM), unless the <b>"Not Destroyable"</b> option has not been enabled in the <b>'Screen Style Properties'</b> .
<b>Open Modal (pop-up screen)</b>	This command consents the opening and displaying of Screen as a 'Modal Window', meaning that it will be managed in the same way as a Dialog Box. There it will open on top of the Screen window from which it was called up. The Modal Screen is a secondary window which captures all the input from the user until it is closed.
<b>Open Frame (multi-monitor)</b>	This command opens and displays a Screen by opening a new window different from Movicon's main window.
<b>Open in other process (Safe Mode)</b>	This command allows the opening and displaying of a Screen executing another instance. (therefore a process in a different memory area). This is useful when Screens using ActiveX might cause an error in the main application. <b>Using this mode however consumes more overall resources whether for the Screen in 'Safe Mode' (which acts as client) and for the one loading it (which acts as server).</b>
<b>Print</b>	This command consents to direct printing (in the printer set as the operating system's default printer) of the selected Screen. In order to get more print options, such as the window used for selecting the printer or print preview, you will need to use the "PrintSynoptic" function from the "ScreenCmdTarget" basic interface.

<b>Close and Return Back</b>	This command closes the selected Screen. If the Screen has been opened in 'Normal' mode, the current Screen will be closed and the previously displayed screen loaded. This happens only when the screen to be closed is the same one displayed on video screen or when nothing has been specified in the "Close" command's "Screen" parameter.
<b>Execute Synapses</b>	This command type activates the Synapses logic contained in the specified Screen. The logic will always be executed in 'asynchronous' mode. The command can also be executed by Basic Script, where it can be executed in 'synchronous' or 'asynchronous' mode.
<b>Open Next (open as normal next ID screen)</b>	This command opens the next screen to the one already opened, based on a map determined by the value set in the ID property of each screen. The next screen will open in "Normal" mode. For further information on how to use this command see "Screen Change commands based on Screen IDs".
<b>Open Prev. (open as normal prev. ID screen)</b>	This command permits you to open previous screen to the one already open, based on a map determined by the value set in the ID property of each screen. The previous screen will open in "Normal" mode. For further information on how to use this command see "Screen Change commands based on Screen IDs".
<b>Capture and Print</b>	<p>This command allows you to capture the open screen and send it to the printer predefined in the operating system in use. This command uses the "PrnWnd.exe" (PrnWndCE.exe per WinCE) tool, which is installed in the same folder in which Movicon is installed. The tool is called automatically by Movicon when the command is executed, and it is Movicon that always passes the handle of the window to be printed as well.</p> <p>For further information on how to use the "PrnWnd.exe" tool see the appropriate "Tools" section.</p>

For further information refer to the paragraph on **"Opening Screen Modalities"**.

#### Monitor

The Monitor where the Screen is to be opened is specified in this parameter. Therefore the system must be correctly configured to manage the Multi-Monitor beforehand.  
The '0' value opens the window on the Monitor set for 'default' in the operating system.



When 'Open Normal (Movicon MDI)' has been selected as **'Action'** the window will be opened on the default Monitor independently of the value set in the **'Monitor'** property.

#### Parameter File

The name, and eventually the path, of the parameter file is entered in this edit box when it is required to open the Screen with parameters.  
For further information on Screens with parameters refer to the paragraph on **"Screens with Parameters"**.



When the file name is entered only and not the path, Movicon will search for the file in the current folder, then in the folder where the Screen is

found, and finally in the folder defined for the Images through the "Images Path" property which can be accessed from the **'Project Path' settings**.



**ATTENTION!** Movicon is a **Unicode** application, therefore the Parameter file must be saved in **Unicode** version.

#### X Pos

The window's origin position (left side) for the X axis is set in this property. The value is expressed in pixels, and the zero point represents the zero point of the monitor's horizontal axis (left side).



When the **'Open Normal (MDI Child)'** has been selected as **'Action'** the window will be opened on the Monitor's zero origin point.

#### Y Pos

The window's position of origin (top side) for the Y axis is set in this property. The value is expressed in pixels, the zero point refers to the Monitor's vertical axis (top side).



When the **'Open Normal (MDI Child)'** has been selected as **'Action'** the window will open on the Monitor's zero point origin.

#### Width

The size of the window's width is set in this property. The value is expressed in pixels.



When the **'Open Normal (Movicon MDI)'** has been selected as **'Action'** the window will open with the sizes set in programming mode.

#### Height

The widow's height is set in this property. The value is expressed in pixels.



When the **'Open Normal (Movicon MDI)'** has been selected as **'Action'** the window will open with the sizes set in programming mode.

#### Caption

When enabled, this property consent the displaying of the title bar for the selected Screen.

#### Border

When enabled, this property gives more highlight to the selected Screen's outer border.

#### Resize Border

When enabled, this property allows the selected Screen to be resized during Runtime. This can be done by using the mouse and the usual **Windows** techniques.

#### Sys Menu

When enabled, this property allows the System Menu to be displayed on the selected Screen's title bar. This setting will have not effect If the **'Title'** bar has not been enabled. The System Menu can be accessed through the icon on the top left in the title bar. Also when enabling this property, the button for closing the window will be made available and displayed on the top right of the title bar (✖).

#### Maximized Box

When enabled, this property allows the button for enlarging the window to be displayed on the top right in the title bar (🔳). If the **'System Menu'** has not been enabled, this setting will have no effect.

#### Minimized Box

When enabled, this property allows the button for reducing the window to be displayed on the top right in the title bar (🔲). If the **'System Menu'** has not been enabled, this setting will have no effect.

## Script Command

---

This group of commands are used for carrying out operations of the Movicon VBA Script.

### Script

The name of the Script VBA to be run or stopped is entered in this edit box (or selected with the '...' browse button on the right).

When you need to select a script from a child project the following syntax should be used:

**ChildProjectName\ScriptName**

When you need to select a script from a Parent project the following syntax should be used:

**..\ScriptName**

### Action

This property allows you to select the action that has to be executed for the selected VBA Script. The choices are:

Action	Description
Run Normal	Run the script in normal mode
Run Synchro	The Basic Script is run synchronized with the component or interface that put it into execution. In this case the object in question will remain blocked until the Basic Script has stopped running.
Run Safe	Runs the script in Safe mode. In this case the script will be executed in a new Movicon process, allowing the Movicon application to be saved if the script causes a crash. Please keep in mind that script loading/unloading is rather slow.
Stop	Stop the execution of the script, but don't unload it from memory.
Unload	Unload the script from memory <i>The unload command of a basic script resource unloads only those basic resources which are being run in separate threads from memory. The other basic script resources being run in the same thread are only stopped. As a consequence of this, the "Unloading" event is no longer executed following an unload command for those basic script resources which are not in separate threads. In addition to this the variables used by the basic script always remain in use once the basic script has been run at least once.</i>

### Parameters

Allows you to set the eventual script parameters. Each parameter must be separated by the "," (comma) character.

**Start,1,500**

The example above is considered as three parameters "Start", "1" and "500". The "GetParameter()" function from the "ScriptMEInterface" should then be used within the Basic Script routine for retrieving the parameters.

### New Instance allowed

This property allows you to run more instances from the same Basic Script at the same time. In order for this to work properly you must, however, specify a number higher than one in the Basic Script's "Maximum Instances" property.

### Timeout

This property allows you to insert a timeout by taking into account when the Basic Script is run in synchronized mode. In any case, the Basic Script will release the interface which called it when the timeout expires.

## User Commands

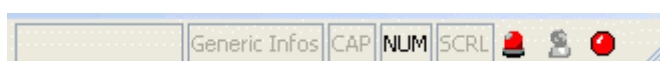
This group of commands allow the operations for Logging on and Logging off Movicon users to be executed.

### Action

The command type or action to be executed or the Basic Script selected is selected through this property. The options are:

Action	Description
<b>Logon</b>	This command opens the window for executing the log on of a user.
<b>Logoff</b>	This commands executes the log off of the user currently active.
<b>Edit Users List</b>	This command allows the editing of users during Runtime to be executed. In this case a executable file is used for editing the XML file containing the " <b>Runtime Users</b> ". The default executable file must be called ' <b>EditUsr.exe</b> ', but the name can be changed by means of the ' <b>EditUsersExe</b> ' registry key. The application name which Movicon must execute upon the edit users in runtime command is saved in this key. When the file name is specified only, Movicon will first search for it in its installation folder then in the Windows list of PATHS.

The Logon and Logoff commands are also available in the Movicon Status Bar:



The icon which represents the user on the bottom right can be in colour or in just grey. Grey means that no users are logged on therefore simply double-click on the icon to display the User logon window where a '0' level authentication will be requested. When the icon is in colour, this means that a user is active and therefore simply double-click on the icon to deactivate the user.

### Level

This edit box is used for setting the lowest level to be associated to the user logging on. If the user does not have any rights for the level specified they will not be authenticated. This property is only significant when '**Log on**' has been selected in the '**Action**' property.

## Report-Recipe Commands

---

This group of commands allows command operations to be executed in the Movicon DataLoggers and Recipes.

### Data Logger/Recipe

The name of the Data Logger or Recipe, where the command selected in the "Action" field is to be executed, is entered in this edit box (or selected by sing the "..." browse button on the right).

### Action

The command or action type to be executed for the selected DataLogger/Recipe. Some of the commands allow reports, which have been previously associated to the DataLogger/Recipe, to be viewed or printed. The Report files supported by Movicon must be created with the editor integrated in Movicon or with '**Crystal Report®**' 10.0 version or subsequent versions.

Action	Description
<b>View</b>	This command executed the opening of a window containing the preview of the Report associated to the DataLogger/Recipe. The Report can also be printed by using the appropriate commands in this window.
<b>Print</b>	This command executes the direct printout of the Report file associated to the DataLogger/Recipe without showing any previews.
<b>View Safe</b>	This command executes the opening of the window containing the preview of the Report associated to the DataLogger/Recipe in safe mode by creating a new process instance. The Report can also be printed from this window by using the appropriate commands.
<b>Print Safe</b>	This command executes the direct printout of the Report file associated to the DataLogger/Recipe in safe mode, without showing any previews, by creating a new process instance.
<b>Move First</b>	This command is only available when a Recipe has been selected in the "DataLogger/Recipe" field. This command allows the values of the first database record to be loaded in the recipe's variable. This is equivalent to the "Move First Variable" variable's function which can be set in the Recipe's " Execution Properties".
<b>Move Last</b>	This command is only available when a Recipe has been selected in the "DataLogger/Recipe" field. This command allows the values of the last database record to be loaded in the recipe's variable. This is equivalent to the "Move Last Variable" variable's function which can be set in the Recipe's " Execution Properties".
<b>Move Prev</b>	This command is only available when a Recipe has been selected in the "DataLogger/Recipe" field. This command allows the values of the previous database record to be loaded in the recipe's variable. This is equivalent to the "Move Prev Variable" variable's function which can be set in the Recipe's " Execution Properties".
<b>Move Next</b>	This command is only available when a Recipe has been selected in the "DataLogger/Recipe" field.

	This command allows the values of the next database record to be loaded in the recipe's variable. This is equivalent to the "Move Next Variable" variable's function which can be set in the Recipe's " Execution Properties".
<b>Activate</b>	This command is available only when a Recipe has been selected in the "DataLogger/Recipe" field. This command executes the activation of the selected recipe. This is equivalent to the "Activate Variable" function which can be set in the Recipe's "Execution Properties".
<b>Save</b>	This command is available only when a Recipe has been selected in the "DataLogger/Recipe" field. This command executes the save of the selected recipe. This is equivalent to the "Save Variable" function which can be set in the Recipe's "Execution Properties".
<b>Delete</b>	This command is available only when a Recipe has been selected in the "DataLogger/Recipe" field. This command executes the delete of the selected recipe. This is equivalent to the "Delete Variable" function which can be set in the Recipe's "Execution Properties".
<b>Requery</b>	This command is available only when a Recipe has been selected in the "DataLogger/Recipe" field. This command executes the requery of the recordset of the selected recipe. This is equivalent to the "ReQuery Variable" function which can be set in the Recipe's "Execution Properties".
<b>Execute Query</b>	This command is available only when a Recipe has been selected in the "DataLogger/Recipe" field. This command executes a query in SQL standard language on data to be selected from the Database. This is equivalent to the "Query Variable" function which can be set in the Recipe's "Execution Properties".

#### **X Pos**

This property is used for entering the horizontal position of Report preview window's origin. The value is expressed in pixels ( '-1' value is the default position).

#### **Y Pos**

This property is used for entering the vertical position of the Report preview window's origin. The value is expressed in pixels ( '- 1 ' value is the default position).

#### **Width**

This property is used for setting the Report preview window's width. The value is expressed in pixels ( '0' value is the default size).

#### **Height**

This property is used for setting the Report preview window's length. The value is expressed in pixels ( ' 0 ' value is the default size).

#### **Toolbar**

This check box is used for displaying or hiding the toolbar in the report preview window.

#### **Group Tree**

This check box is used for displaying or hiding the group Tree in the report preview window.



## Menu Commands

---

This group of commands is used for carrying out operations for displaying Movicon Menus.

### **Menu**

The name of the Menu to be displayed is entered in this edit box (or selected with the '...' browse button on the right).

When selecting a child project Menu the syntax is:

**ChildProjectName\MenuName**

When selecting a parent project Menu from a child project the syntax is:

**..\MenuName**

### **X Pos**

This property is used for entering the horizontal position of the Menu window's origin. The value is expressed in pixels (the '-1' value takes the position of the mouse).

### **Y Pos**

This property is used for entering the vertical position of the Menu window's origin. The value is expressed in pixels (the '- 1' value takes the position of the mouse).

## System Commands

This command group allows operations to be executed for closing Movicon or starting up external applications.

### Action

The command type or action to be executed is selected through this property. The options are:

Action	Description
<b>Shut Down OS</b>	This command executes the shut down of Movicon and the Operating System.
<b>Shut Down App</b>	This command executes the shut down of Movicon.
<b>Launch App</b>	This command launches a Windows application, which must then be specified in the ' <b>Command Line</b> ' property.
<b>Launch App and Wait</b>	This command launches a Windows application, which must then be specified in the ' <b>Command Line</b> ' property.
<b>Play Sound File (Parameter is the Sound file)</b>	allows a wav. sound file to be reproduced by exploiting the preset audio card. The name of the sound file to be reproduced must be entered in the ' <b>Command Line</b> ' parameter.
<b>Beep (Parameter is the frequency of the sound)</b>	Allows a beep sound to be made by using the computer's speaker. The sound frequency must be entered in the ' <b>Command Line</b> ' parameter (ex. values from '37' to '37767' are permitted)The beep duration time is entered in the "Timeout" parameter in milliseconds. Only one sound can be used in WinCE.
<b>Speak (Parameter is text to speak, not available for WinCE)</b>	Allows the text, set in the ' <b>Command Line</b> ', to be executed in text-to-speech mode. This command supports multilingual for which you can also set string IDs from the project string table: Not available on WinCE.
<b>Reboot OS (not available for WinCE)</b>	Allows you to command a reboot of the operating system. However, to monitor the status of an application it would be best to use the service and the "Recovery" (Service Property, recovery folder) just in case you have to intervene and take action when errors occur. Not available in WinCE.

### Command Line

The command line for executing a Windows application is entered in this edit box. If the executable file path is not specified, Movicon will search for it in the Windows' folders.

### Timeout

The timeout, which Movicon attends to when application is not responding, is specified in this edit box in milliseconds.

## Language Commands

---

This group of commands are used for carrying out operations to change languages in Movicon texts.

### Language

The language to be activated is selected through this property. The languages set in the **Strings Table** are listed. If this field is left blank, therefore without a selected language, the list of available languages in the project will open up in a dialog window when executing the command. Select the language desired from this list and activate it by confirming with OK.

## Help Command

---

This group of commands are used to execute the display of help or text message files.

### Action

This property is used for selecting the command or action type to be executed. The choices are:

Action	Description
<b>Topic</b>	This command executes the opening of the topic specified in the <b>'Topic'</b> property. The topic will be searched for in the help file associated to the project, by means of the "HTML Help File" property available in the <b>'Project Work Folder Path Settings'</b> .
<b>Tooltip Popup</b>	This command displays a pop-up message. The text has to be inserted in the <b>'Topic'</b> property and can also be as string ID existing in the project's String Table.

### Topic

The topic name or the pop-up text to be displayed is entered in this edit box. This property changes according to the selection made in the **'Action'** property.

## Event Commands

---

This group of commands consents Events to be executed no matter what their activation conditions are. This functionality consents the project's Events resource to be used as a container for commands to be used in the project. By keeping the commands list in one unique point, will make it possible to edit just one command list of one Event object in order to automatically modify the command lists of all the object referring to that Event object.

Any activation conditions associated to Event objects will remain valid.

### Event

The event to be executed can be selected in this property. All the Event objects in the project's "Event Object List" will be included in the list.

## Alarm Commands

This group of commands are used for carrying out the acknowledge and reset operations of the Movicon alarms.



The 'Period', 'Duration' and 'Date' parameters are optional. If one or all of these parameters are not inserted they will be requested through an appropriate dialog window configured in the Report when doing a preview or a direct print. Otherwise all the commands will be indicated without showing a dialog window and the Report will use these parameters to filter the data.

### Action

The command or action type to be executed on the Movicon alarms is selected through this property. The choices are:

Action	Description
<b>Ack All</b>	This command executes the acknowledgement of all the active alarms.
<b>Reset All</b>	This command executes the reset of all the active alarms.
<b>Sound ON/OFF</b>	This command enables or disables the management of the acoustic signal which will be activated for unacknowledged alarms.
<b>Show Report</b>	Opens a Report preview window of the alarms. This command is only available when the "Alarm Statistics" option has been enabled on the license.
<b>Print Report</b>	Directly sends the alarm Report to be printed without opening a preview window. This command is only available when the "Alarm Statistics" option has been enabled on the license.
<b>Export Report</b>	This command allows you to export alarm statistic reports to html files. The exported file is saved in the project's "DLOGGERS" folder in the name of "<Report Name>_<gg/mm/yy>.html". This command is only available when the "Alarm Statistics" option has been enabled on the license.

### Report File

Movicon supports two commands designed for interacting with the alarm reports. The commands are 'Show Report' and 'Print Report'.

Each command permits you to choose which report file to open or print in the 'Report File' parameter. The drop-down list reports the names of four existing and already available Reports, but other names can also be entered. If the customized report to be displayed or printed is in the Movicon installation folder then only the report name needs to be entered without the extension (ie. MyReport); otherwise the whole path including the extension will have to be entered (ie. C:\Temp\MyReport.rpt).

The four default reports have been created with the Crystal Report 10 version and are interfaced with the 'Alarms' table from the Movicon Historical Log:

1. **OrderByDate.rpt:** Report of alarms in activation date order. Each alarm is reported with the date and time of its activation, the date and time of its occurrence and duration. The total of all the durations throughout each day and the selected period are represented. The last page reports a bar chart with the number of alarm appearances each day
2. **OrderByDuration.rpt:** Report of alarms in duration order (starting with the longest duration to the shortest). Each alarm is reported with the date and time

of their activation, the data and time of their occurrence and duration. The duration totals of the period selected is represented. The last page reports a line chart showing the longest, average and total of durations in seconds of each day

3. **GroupByThreshold.rpt:** Report of single alarms in alarm groups. Each alarm is reported with its description, date and time of activation, date and time of occurrence and duration. The total duration of each alarm for the period selected and the total duration of all the alarms in the period selected are represented. The last page reports a pie graph with the duration percentages of each single alarm for the period selected
4. **GroupByFrequency.rpt:** Report of alarms grouped by frequency. Each alarm is reported with its description and how many times they occurred in the period selected. A classification of the first three most frequent alarms in the period selected is also represented. The last page reports a pie graph with the frequencies of each single alarm in the period selected

Movicon searches for these Report files in the same folder where Movicon.exe. is found.

#### **ToolBar**

This property permits the toolbar to be displayed or hidden in the report preview window.

#### **Group Tree**

This property permits the group in tree structure to be displayed or hidden in the report preview window.

#### **Period**

This property permits a temporal filter to be set for extracting data from the database. The possible values are:

Selected date time

- Today
- Yesterday or today
- Current week
- Current month
- Current year
- Last 7 days
- Last 30 days
- Last 60 days
- Last 90 days
- Last 1 years
- Last 2 years
- Last 5 years
- Last 10 years

The filter is carried out according to the activation date of each alarm.

#### **Duration**

This property permits a filter to be set on the duration of each alarm. The default value is '00:00:00' but the filter can be set so that only the alarms which last longer than a certain set time in 'hour:minutes:seconds' are retrieved from the database.

#### **Date**

This property permits a filter to be set on the date. This setting is only valid when the 'Period' parameters have been set in the 'Select date time'. Two dates can be inserted here for representing the start and end date for retrieving data. The format is: "dd/mm/yyyy hh:mm:ss dd/mm/yyyy hh:mm:ss".

# 18. Alarms

***The Movicon Alarms resource consents the managing of diagnostics and messages in projects.***

The diagnostics is one of the fundamental elements on which any supervision application project is based. Basically, the system's task is to generate alarms in relation to variable or logic intervention thresholds.

The Movicon management complies with the ISA (S-18) normative, but is completely configurable to adapt to any application requirement. The project alarms are grouped in objects which are then grouped into thresholds.

Alarm activation can be evoked by a fixed threshold value or by another tag.

The Acknowledge or reset operations on each alarm can be configured, just like the style and functioning modalities can be. The priority levels, which can be assigned to each alarm by the programmer, are virtually unlimited. The Alarms can be grouped into 'Areas' to obtain display filters for the operator or Alarm Window dedicated to each plant sector. The time-stamping is precise and accurate to the millisecond. Movicon also integrates calculated statistics of the alarms with the highest frequency and/or duration in the specified period. Managing alarms in network architectures has been extremely simplified by connecting alarm windows to network stations, or to OPC AE servers.

The alarms also support tag-linked texts, therefore the real values measured at alarm interventions can be displayed and recorded. The alarms also manage SMS, Fax, Email, Speech notifications, through customizable sound files and any kind of intervention, acknowledgement and reset customizing can also be done through the VBA script in response to each alarm event.



The Movicon alarms can be configured so that they can be acknowledged and reset by the operator before disappearing or they can be linked to the status of a variable only, becoming just simple notification messages.

The "**Alarm Window**" window, which can be inserted in any of the project's Screens, is used for viewing the Alarms' situation. The operator can verify and interact with the alarms' situation through this window, evoking the acknowledge or reset commands according to the ISA level 2 norms.

The Alarms can be grouped in one certain area, to allow the Alarm Window to display the alarms relating to a specific logic area of the plant only.



An Alarm Window can be dedicated to display the remote alarms of a station (Server) connect in net. through the Networking functionalities.

The displaying of Alarm Window is setup by the project's developer by configuring the window in order to adapt the project to the plant's graphical or functional requirements.

## **Alarms List Resource**

---

Movicon has a powerful tool for configuring the alarm objects to be managed in the project. The '**Alarms List**' is the tool for generating the events which will be represented in the appropriate display windows placed in the Screens.

Each single alarm is considered as an object, which can be completely configured through the '**Properties window**'. The property of each single alarm allows you to completely customize both the thresholds and execution functionalities.

The Alarm objects, after being configured, can be saved in the Symbols Library, as Templates. Each single alarm can be saved in the library individually or associated to the symbol required.

This technology, defined as Power Template, permits the programmer to drastically reduce project developing time, by allowing them to insert symbols, automatically inserting variables and other Templates linked to them.

The '**Alarms List**' resource is available in the Movicon '**Project Explorer**' window.

## **Scroll Lock Led Key**

---

The "**BlinkLedKeyboard**" register key can be configured so that any alarm presence still not acknowledged causes the '**Scroll Lock**' key to blink on the keyboard. This enables the presence of new alarms to also be seen at a distance.

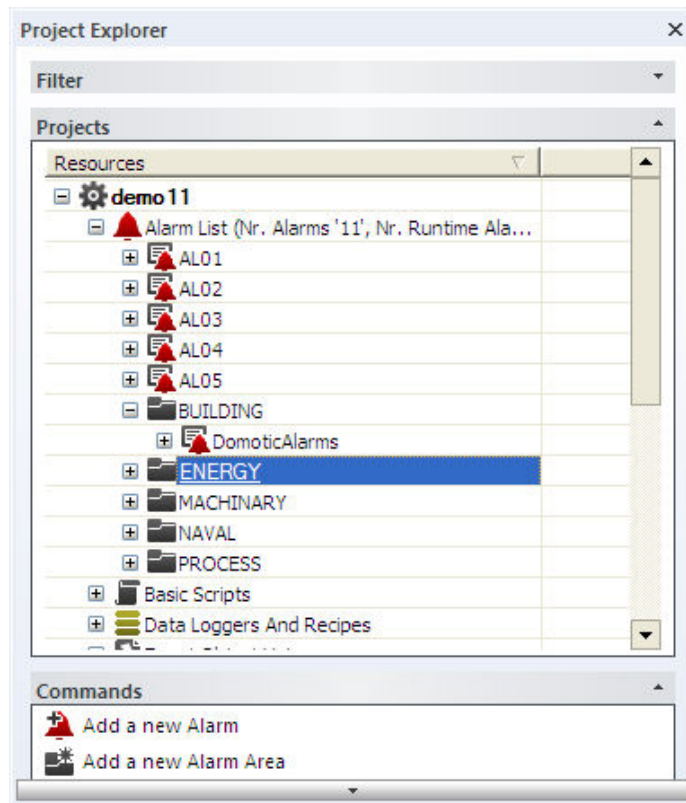
## **Alarm Printout Spooler**

---

Movicon manages the printout of alarms and events directly to the printer as they occurs. Please refers to the Project Spooler Manager Settings properties.

# Inserting Alarms

To insert an Alarm into the project, you need to start this procedure by inserting a new object into the 'Alarms List' group in the 'Project Explorer' window. To do this you can either right mouse click on the 'Alarms List' group in the 'Project Explorer' window and select the 'New Alarm' command, or use the same command also found in the Project Explorer's 'Commands' window.



When confirming this operation, the new alarm will appear in the group or in the point selected in the project's structure. At this point you can go on and carry out the Alarm property settings as described in the documents on **"Alarm Properties"**.

Afterwards the Alarm object can be assigned a Name by clicking on the resource and entering the name replacing the one for default, or after having selected the object, press the F2 key and carry out inserting the new name procedure.

New Alarms can be entered by copying the data from the resource, from the same or other projects. In order to do this, select the alarm or the alarms you wish to copy, then use the Copy/Paste function from the Edit menu (or with the equivalent keyboard keys, tool bar or right mouse key techniques).

To delete one or more alarms from the project first select them then activate the 'CANC' key or 'DELETE' key from the keyboard.

The setting or editing of alarms is done through the Movicon 'Properties Window'.

Each entered alarm must then be associated with one or more intervention thresholds. To add an intervention threshold to an alarm, first select the alarm then right mouse click and select the 'New Alarm Threshold' command, or use the same command which can also be found in the Project Explorer's 'Commands' window. When confirming this operation, the new threshold will appear in the Alarm's structure. At this point you can proceed with setting the threshold's properties as described in the document about **"Alarm Properties"**.



## Messages

---

The only difference between alarms and messages are the "Support ACK" and "Support RESET" ' property settings of the alarm threshold. If these two properties are both disabled, the alarm will not need acknowledging and resetting by the user and therefore will only be displayed according to the status of the associated variable, thus making it a simple display message.

In the Project's Explorer Window 'Alarms List' the image displayed for the alarm thresholds will differ according to whether they have been configured as alarms or as messages.

During runtime the image shown in the Alarms Window, next to the text, will also differ according to whether an alarm or message is being dealt with. This function is valid for local alarms only. Different icons, to represent whether an alarm or message, are not supported for alarms and messages displayed by Client projects. In addition to this, "MESSAGE ON" and "MESSAGE OFF " instead of "ALARM ON" and "ALARM OFF" texts will display as the event identifier in the "Historical Log Window". Therefore the MESSAGE ON" and "MESSAGE OFF " events can be filtered in the Historical Log's message filter window.

## Alarm Areas

---

The alarms can be grouped in different areas so that they can be displayed in the "**Alarm Window**" window based on their on the area they belong to. This may be helpful to enable you to divide the alarms according to the different plant zones they come from. There are two methods to use for associating an alarm to a certain area:

1. insert the name of the area in the "**Area**" property in the "Alarm Threshold General Properties". This will associated the Area name to the alarm threshold.
2. create Areas in the "Alarm List" resource by using the "Add new Alarm Area..." command. This command allows you to add folders to the "Alarm List" resource. Each folder represents an area, and all the alarms which are moved to this folder will then belong to that specified area.



Warning! When both methods are applied to the same alarm, priority will be given to the Area set in the "Alarm Threshold General Settings".

## Importing Alarms from other Projects

---

Movicon allows you to copy one or more Alarms from one project to another. To do this, both projects must first be opened, then the Alarms selected and copied using the Copy command from the 'Project Explorer' window of the source project. Then position mouse pointer in 'Project Explorer' window of the destination project and execute the Paste command. The copied Alarms should then also be available in the destination project.

The Drag & Drop techniques can also be used with the following procedure: select the Alarms from the source project and keep the left mouse key pressed and drag them to the point desired in the destination project, then release the mouse key.

## Enabling Alarms

---

The Movicon Alarms predisposed with a useful enabling property, to allow the programmer to temporary deactivate the working of each single alarm.

This setting can be done by accessing the "**Enable**" item through the '**Alarms General Properties**' of the Alarms' 'Properties Window'.

Alarm groups or the all the contents of the Alarms Editor can also be enabled or disabled at the same time. In this case simply select all the alarms required from the 'Alarms List' and set the "**Enable**" property from the '**Alarms General Properties**'. In this way the setting will be propagated to all the alarms selected.

# Alarms Historical

---

The alarms have two pieces of useful information to give them more detail, a **Unique ID** and a **Transaction ID**. The Unique ID is created the once the project is put into runtime mode for the first time, and is a number between 1 and 2147483647, being a Long number. This value is unique for all the alarms, meaning that each alarm has its own Unique ID, which is saved in the alarms' status file in the project's ALARM folder. Once this ID has been created it will remain so until the alarm's status file is cancelled after which it will be assigned a new Unique ID which may be different from its last one. The Transaction ID, however, is increased each time the alarm turns ON and remains at this value until the alarm turns ON again. Therefore all the ON,ACK, RESET and OFF events are recorded with the same Transaction ID. The Transaction ID's initial value is zero. Its value is saved in the alarm's status file and is loaded at the project startup. The value limit is 2147483647, long number type value that once reached restarts from the 1 value. The Transaction ID restarts from zero when the alarm's status file is cancelled.

These two ID values are recorded in the Historical Log table as well and where two new columns have been inserted, "UnIID" and "TraID" for this purpose. The values of these two columns only have meaning for the Historical Log "Alarm" table and not the "SysMsgs" and "Drivers" table.



**After entering these new features, in a Client-Server configuration all the applications must be aligned with the same Movicon 11 version. In fact an Alarm Window or a Historical Log window will not be able to retrieve logged alarms or events from a project run with an earlier Movicon version.**

Each time an alarm event is recorded, it will be identified with an Unique ID and a Transaction ID. This consents to a more structured and detailed alarm information in the "Alarms Window and in the "Historical Log Window":

## Historical Log Window

The alarm events are displayed in groups according to the Transaction ID. For instance, the "ON", "ACK", "OFF" and "RESET" events recorded for the same alarm event will have the same Transaction ID, and the one same Unique ID, and will be displayed as an Alarm subgroup. There is a "+" symbol next to the alarm's text that when clicked on opens a tree structure showing all the events of that alarm.

## Alarms Window

The alarms are listed in the window in the usual way. However, when selecting an alarm and clicking on the "Get History" button, all the history for that alarm will be retrieved, which included all the times the alarm was activated and the events generated for each activation ("ON", "ACK", "OFF" and "RESET"). A "+" symbol will appear on the left of the alarm's text which will open a tree of the alarm's chronology when clicked on.

## Alarm Statistics

---

There has also been a series of function added to the "AlarmThresholdCmdTarget" interface which allow you to get statistical information on alarms. This information is always saved in the alarm's status file:

**TotalTimeOn:** returns the alarm's total on time

**LastTotalTimeOn:** returns the date and time of the alarm's last ON transaction

**LastComment:** Sets or returns an alarm comment

**GetTotNumOn:** returns the total number of times the alarm went ON

**GetTotNumAck:** returns the total number of times the alarm was acknowledged

**GetTotNumReset:** returns the total number of times the alarm was reset

**GetUniqueID:** returns the alarm's Unique ID

**GetTransactionID:** returns the alarm's transaction ID

# Alarms as Templates

---

Movicon has a very handy functionality which can be used when the project has many alarms with similar text messages and are of the same type associated to a series of different variables.

- **A common example:** A project must manage 200 motors which all have "Motor Overload" alarm conditions. The RealTime Database has 200 variables exchanged with the field relating to the motors' status. These variables have to generate 200 different alarms, one for each motor, but the alarms are actually all of the same type except for the name of the motor on each alarm and the associated variable.

In cases such as above, you need to have a tool which parameterizes the generation of the alarms, providing you with the possibility to **setup the alarm only once** and quickly associate it to the 200 variables, such as in the case above. In Movicon, this functionality is called **"Alarm Template"**.

To obtain this useful function you need to define the alarm 'type' (Template) in the Alarms Resource. Associate the desired intervention threshold (or thresholds) and the relating "standard" text (the text used in the above example is "Motor Overload"). Then you can select the variables from the RealTime Database's Variable List and use the 'Associate an Alarm' command with the right mouse key to select the desired alarms from the alarm list. This will link the variables (in our example this would be the 200 motor variables) to the alarm 'Template'. Therefore with one click on the alarm Template Movicon will automatically setup 200 different alarms in the project. Each variable will be monitored to activate an alarm whenever any threshold is exceeded. The displayed alarm text will show the standard text (in our example above this would be "Motor Overload") and the name of the associated variable.

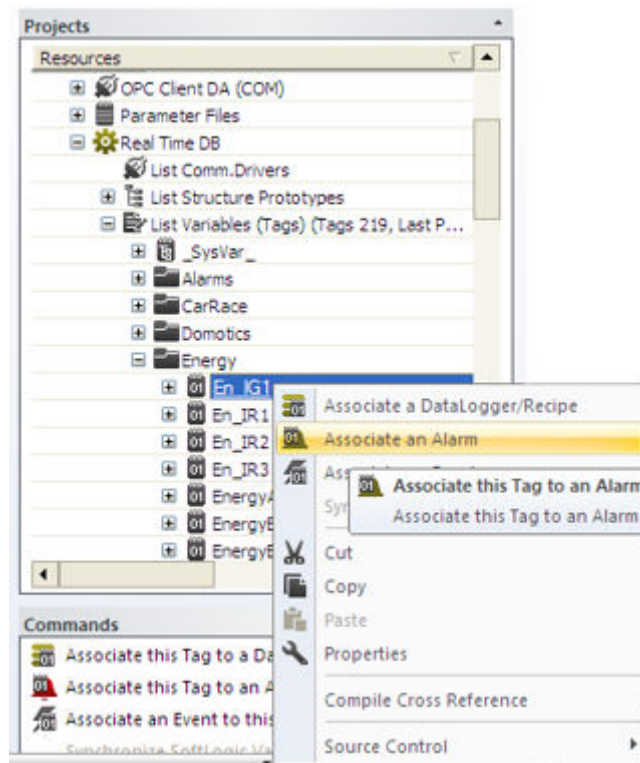
## Example:

1. Create a new alarm object with a standard alarm text with "Motor Overload". Set an intervention threshold.
2. Associate the alarm created as a Template to 200 variables existing in the project with names from MT1 to MT200. This association is done by selecting the group of variables indicated (Shift + click or CTRL + click) and using the right mouse key with the "Associate an Alarm" command.
3. The assigned alarm icon will appear in all the variables from MT1 to MT200.
4. When a threshold is exceeded during runtime, for instance the MT33 variable threshold, an alarm will appear indicating "MT33 - Motor Overload".



The name of the variable inserted as "standard" text can be replaced with a customized text. To do this, just insert a string in the "String Table" with an ID in the same name of the variable. When the alarm is activated, it's name will be replaced with the text contained in the string.

You can associate an Alarm to a Variable by right clicking on the Variable in the "Variable Group" list presented in the "Project Explorer" window and then select the "Associate an Alarm" command, or use the same command also found in the Project Explorer's "Command window".



The request to associate and alarm opens another window containing the list of alarms which were inserted beforehand and are now available for use.



In the properties of the Alarm to be associated to a variable, you should not specify any reference variable but only the text and activation condition.

## Alarms activated by homonymous Variables

When an alarm is not associated to an activation variable, "Alarm Variable" property, Movicon will control if a variable exists in the Real Time DB with the same alarm name in runtime. If a variable does exists with the same name, it will be used as the alarm's activation variable and therefore its value will be used for activating the alarm based on its threshold activation settings.

For instance, if an alarm has been defined as "Alarm01" without an activation variable associated to it and a variable called "Alarm01" exists in the Real Time DB, this variable will be used by the alarm as its activation variable.

In this case the alarm can also be associated as a template to other variables and result as a different alarm for each variable, one for the variable which has the same alarm name and one for each variable to which the alarm has been associated as a template.



**When using alarms as templates, therefore without any associated activation variable, be careful not to define variables in the Real Time DB with the same name as the alarm's, otherwise a link to the variable with the same name of the alarm will be created in addition to the alarms as templates as well.**

# Alarm Status File

---

Each alarm defined in the project will create a Status File in XML format in the "ALARM" folder during runtime mode. This file is normally created with a syntax of this type:

ProjectName\_ThresholdNameSoglia\_VariableName.alr

The reason for having this file, as for the variable retentivity file, is to keep saved certain information inherent to the alarms when the project is closed. The saved information gives details on the following:

- **Alarm Status:** this lets you know at the project startup which status, ON, OFF, ACK etc, the Alarm was when the project was last terminated. This serves for restoring the present situation back to how it was before the project was last terminated
- **Total Time ON:** shows the total time in which the alarm remained in the ON status
- **Last Time ON:** shows the date and time of the alarm's last ON transaction
- **Comment:** shows any comments associated to the alarm by users by using the appropriate script functions
- **Number of ON Events:** shows how many times the alarm turned ON
- **Number of ACK Events:** shows how many times the alarm was acknowledged
- **Number of RESET Events:** shows how many times the alarm was reset
- **Unique ID:** shows the alarm's unique ID
- **Transaction ID:** shows the alarm's transaction ID

Some of this information such as the Unique ID and the Transaction ID is important for identifying that particular alarm for managing the "Alarms History" in the "Alarms Window" and in the "Historical Log Window". The other information is just for statistical purposes.



**Cancelling an alarm Status File will loose all the statistical information and identifiers for that alarm. A new file will, however be automatically recreated by Movicon again.**

## Alarm Properties

---

The alarms and messages inserted in the 'Alarms List' can be completely customized in the properties. The alarms and messages are built from intervention thresholds, each one composes the alarm structure whose settings can be edited in the '**Properties window**'.

### Alarm General Properties

---

The moment an alarm or message is inserted it can be configured in the general properties as described below. The general properties allow you to associate the variable which determines the alarm's intervention.

To modify an alarm's general settings, select it with the mouse and use the 'Movicon '**Properties window**'.

#### Name

This edit box permits you to define the Alarm or message object's name which is to be configured.

The object's name only identifies the alarm or message in the '**Alarms List**' and identifies the object if inserted into the Template library.

### Device Name

This edit box allows you to define the device's name that the Alarm or Message refers to. The device's name will be displayed in the alarms window, before the alarms' text. This will allow you to manage the alarms as Templates, keeping the same alarms text referred to different devices.

For instance, the 'Overheating' alarm text can be referred to the device of interest, by modifying it each time the template is inserted (eg. Mot.1, Mot.2 etc.). The text which appears will then be determined by <threshold name> - <device name> : <alarm text>.



The variable value can be displayed in the text by inserting the following syntax:

%(Variable Name)

For instance, if you want to insert the VAR00001 value in the text, the string should be:

"Device Name %(VAR00001)

By doing this the variable's value will be displayed dynamically. The value displayed in the Alarm Window will be the value read the exact moment the alarm was turned ON.

### Alarm Variable

This box is used to select the variable from those presented in the Movicon 'Variables List' which generate the alarm or alarms (or messages) in function with event thresholds, defined through the relevant properties. You can use the syntaxes relating to the bit pointing within a variable or structure variables (eg. VAR0001.5, or STR0001:ALL\_01).



If this field is left blank, in runtime Movicon will control if there is an existing variable with the same name of the alarm in the Real Time DB. If one does exist it will be used as the activation variable for that alarm (see the section on "Alarms activated by homonymous Variables").

### Var. Duration

Enabling this property will make the alarm work with the usual threshold, but the confrontation will be referred to the "Total Time ON" instead of variable's value. The time is the total duration expressed in seconds that the variable was at a value different from zero. By using a new variable statistics (see "Retentive Variables and Statistical Data") you will be able to find out how long the variable has been at a value that is not zero. This function can be very handy in managing programmed maintenance. In order to use this function you will need to enable the "Retentive not Shared" and "Enable Statistic Data" properties for the variable in question.

When the variable's "Total Time ON" exceeds the value set in the alarm's threshold, the alarm will be activated. After which by resetting the variable's statistics, its "Total Time ON" will be zeroed and the alarm will turn to OFF.

### Enable Alarm Variable

This box is used for selecting a variable which consents the alarm intervention. When the selected variable has a value other than zero (><0), when conditions subsist, the alarm will be notified regularly. When the variable has a 0 value, even though there maybe alarm conditions present, this will not be activated.

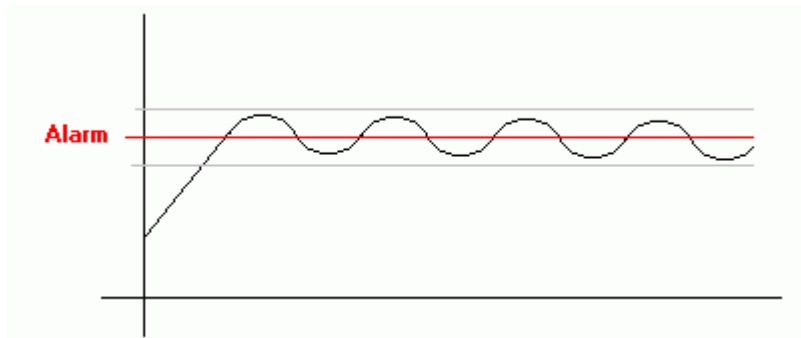
When this box is not selected, the alarm will always be enabled if the 'Enable' property is set at 'True' value.

### Enable Dispatching Messages Var.

This selection box is used for selecting a variable to consent the sending of messages to the Alarm Dispatcher. The sending of sms, email,etc. messages to the Alarm Dispatcher will be enabled when the selected variable is set with a value other than zero (><0). On the other hand, when the variable is set at zero value the sending of messages to the Alarm Dispatcher will be disabled.

### Hysteresis Alarm Value

The alarm's hysteresis allows you to insert a control on the alarms effective re-entry when the threshold value is oscillating, in order to provoke an over-intervention (ON) and re-entry (OFF) of the alarm itself.



The hysteresis is an absolute value which is dealt differently according to how the condition of the alarm's intervention is set:

**Condition "="**

The alarm intervenes when the variable assumes a value equal to the value set for the alarm's threshold. The alarm re-enters when the value of the variable is higher or lower than the threshold value  $\pm$  half of the hysteresis value.

**Condition ">="**

The alarm intervenes when the variable assumes a value higher or equal to that of the alarm's threshold. The alarm re-enters when the variable is lower than the threshold value less the hysteresis value.

**Condition "<="**

The alarm intervenes when the variable assumes a value less or equal to the that of the alarm's threshold. The alarm re-enters when the variable is higher than the threshold value plus the hysteresis value.

**Exclusive Threshold**

When there are more than one threshold present, the enabling of this property will evoke the disappearance of a alarm referred to one threshold when the alarm referring to the next threshold appears. Otherwise both alarms will both remain active.

**Enable Alarm Variable**

This selection box is used to enable or disable the alarm intervention.



The **'Enable'** property has priority over the **'Enable Variable'**.

**Enable only if Quality Good**

This selection box is used for enabling the alarms management only when the quality of the its connected variable is good. This means for example that the alarms associated to the variables directly arriving from the PLC or from any other field device, will automatically be disabled when there are any communication interruptions.

## Alarm Threshold General Properties

The alarms and messages inserted into the 'Alarms List' are equipped to have one or more intervention thresholds according to the functions to be carried out. The threshold interventions determine the alarm's or message's behaviour when the defined threshold values are reached or exceeded. The Threshold properties can be activated in the editor through the tree structure of each single Alarm.

To modify the general settings of the alarm's thresholds, select the threshold with the mouse and use the Movicon **'Properties Window'**.

**Threshold Name**

The name of the threshold to be configured is defined in this edit box. The threshold's name will be displayed in the alarms window, preceding the alarm's text.

This allows you to manage the alarms as Templates keeping the same text referred to different devices.

For instance, the 'Overheating' alarm text can be referred to the device of interest, by modifying it each time the template is inserted (eg. Mot.1, Mot.2 etc.). The text which appears will then be determined by <threshold name> - <device name> : <alarm text>.

#### Alarm Area

This box is used to associate one or more areas for the Alarms or Messages in question if needed.

This way the alarm will be shown activating the selected area.

#### Alarm Text

The threshold's Title string is typed in this box which will constitute the alarm's text. You can select the text among those inserted in the project's String resource. In this case the text may be subject to the Language Change functionalities.

The text which appears in the Alarms or Messages window will be determined by <threshold name> - <device name> : <alarm text>



The variable value can be displayed in the text by inserting the following syntax:

%(Variable Name)

For instance, if you want to insert the VAR00001 value in the text, the string should be:

"Alarm Title %(VAR00001)"

By doing this the variable's value will be displayed dynamically. The value displayed in the Alarm Window will be the value read the exact moment the alarm was turned ON.

#### Alarm Help

This box, whose use is not obligatory, allows you to type the text string which will constitute the eventual alarms help guide. The Help guide will be available to the operator during runtime by double-clicking on the alarm or activating the appropriated information button. The box allows you to select the text from those inserted in the project's string resource. In this case the text may be subject to the Language Change functionalities.

#### Duration Message Format

The message to be filed in the 'CommCol' column of the Historical Log's 'Alarms' table is entered in this property. The message is recorded only on an event of 'Alarms Off'.

The message may only include the following special codes:

- **%H** = Duration of alarm in hours
- **%M** = Duration of alarm in minutes
- **%S** = Duration of alarm in seconds

If this property is left empty, Movicon will automatically insert the total alarm duration with the following string:

**Total Duration 00:00:00**

where 00:00:00 indicates the alarm's duration in hours, minutes and seconds.

#### Read Access Area Level

By using this property you can setup the Access Level mask needed for displaying the alarm in the "Alarm Window". When the Access Level mask of the user currently logged on does not correspond to the control's settings, the user will not be able to see the alarms. The "0000" and "FFFF" settings make the object accessible in read to any user.

The historical logging operations of the alarms are carried out independently of the access rights of the user logged on at that moment.

For further details on 'Access Masks' please refer to the **"User Levels and Access Levels"** paragraph.

#### Write Access Area Level

By using this property you can set the Access Level mask needed for executing, for instance, an alarm acknowledgement and reset. When the Access Level mask of the user currently logged on does not correspond to the control's settings, the user will not be able to execute any command operations associated to the alarms. The "FFFF" level makes the object accessible in write to any user.



For further details on 'Access Masks' please refer to the **"User Levels and Access Levels"** paragraph.

## Alarm Threshold Execution Properties

---

The alarm's value and intervention condition are defined in the Alarm Threshold's Execution property.

To edit the alarm threshold's executions select the threshold with the mouse and use the Movicon **'Properties Window'**.

### Activation Value

This edit box allows you to set the threshold value to be reached so that the relative alarm be activated effectively. The maximum value permitted depends on the variable type associated to the alarm. You can also associate a Real Time DB variable to this property to make the intervention threshold dynamic during the project run.

### Alarm Threshold Variable

The name of the variable to be used in place of the Value property is entered in this edit box (or selected with the '...' browse button on the right). In this way the threshold value can be made dynamic regarding how much of the variables' contents can be taken and edited during Runtime.

### Activation Condition

This property determines on what condition the alarm is to activate:

- '>=' means that the alarm activated when the variable value is equal or higher than the 'Value' property setting
- '<=' means that alarm activates when the variable value is equal or lower than the 'Value' property setting
- '=' means that the alarm activates when the variable value is the same as the 'Value' property setting
- **"Rate Change Decrease"**: means that the alarm is active when the associated variable's value does not change to the quantity desired ("Activation Value" property) within the time set ("Delay (sec.)" property). The time count starts the moment the alarm is enabled with the "Enable Alarm Variable"
- **"Rate Change Increase"**: means that the alarm activates when the associated variable's value does not change to the quantity desired (Activation Value" property) within the time set ("Delay (sec.)" property). The time count starts the moment the alarm is enabled with the "Enable Alarm Variable"

### Rate change increase/decrease conditions

These two alarm activation conditions can be used for generating an alarm if the associated variable's value does not change to a greater or smaller value within a certain time compared to the one set. These conditions must be used together with the "Delay (sec.)" and "Enable Alarm variable" properties. The function of these conditions is: the moment an alarm is enabled with the "Enable Alarm Variable" the variable's value is recorded. At this point onwards a confrontation takes place with the variable's recorded value and its current one and if the current value has not increased (or decreased) by a greater (or smaller) quantity of the threshold value within the delayed time set, the alarm will activate. Each time the difference between the value saved for the comparison and the actual value of the variable exceeds the set threshold, the comparison value is updated with the current value and the delay time is zeroed and restarts counting from the beginning.

For instance, let's suppose that we have an alarm configured in this way:

Alarm Variable = VAR0001  
Enable Alarm Variable = EnableAlarm  
Activation Value = 5  
Activation Condition = Rate change increase  
Delay (sec.) = 10

At this point when an alarm is enabled by setting the "EnableAlarm" variable to the value '1', the current variable's value is saved as the reference value and the time count starts.

If the value of the variable associated to the "VAR00001" alarm does not increase to a value higher than the set threshold (5) compared to the saved reference value within 10 seconds (Delay (sec.)), the alarm will activate. If the variable increases to a value higher than 5 before the 10 seconds are up, this value is saved as the new reference value and the count is zeroed, after which the control starts from the beginning, meaning that the variable's set value will have to increase above the new reference value before the delay time expires.

This works in the same way when the "Activation Condition" is set with "Rate change decrease": in this case the variable decrease in value is controlled and not its increase.

### **Severity**

This box allows you to select the alarm priority required. The priority will be displayed and recorded in the appropriate field reserved for the alarm.

Is possible to assign a number comprised between 0 and 65535. The alarm window and the Log will record the number of priorities assigned.

### **Delay (sec)**

You can enter a numeric value comprised between 0 and 65535 in this edit box. The value entered, expressed in seconds, will establish the alarm's intervention delay time, thus creating a filter in the threshold.

The default value is zero (no delay).

### **Commands on CTRL + Dbl Click**

This command opens the Movicon "**Command List**" window through which you can set a list of one or more commands to be executed on operator request when the alarm is active. The Command List can be executed, when the alarm is active and displayed in the Alarm Window, by double clicking the alarm while keeping the CTRL key pressed down at the same time.

For further information on the commands available please refer to the paragraph on "**Command List**".

### **Commands when Alarm ON/ACK/RESET/OFF**

This button opens the Movicon "**Command List**" window, through which you can set a list of one or more commands which will be executed by Movicon when the alarms generate the specified event (Alarm ON, Alarm OFF, etc.).

For further information about available commands please consult the paragraph on "**Command List**".

## **Alarm Threshold Style Properties**

---

The Alarm Threshold Style Properties allow you to define the graphic and operating styles. To modify the Alarm Threshold Style settings, select the threshold the mouse and use the Movicon '**Properties Window**'.



By means of the 'Support ACK' and 'Support RESET' properties you can define whether the operator is needed to acknowledge and reset the alarms. When these two properties are disabled the alarms will be displayed according to the associated variable status only, and therefore a simple display Message.

### **Support ACK**

This selection box permits you to determine whether you want the alarm acknowledgement (ACK) activated or not.



When the "Support ACK" property is enabled with the "Support RESET" property disabled, once the alarm has been acknowledged it will disappear from the Alarm Window even though its status is still active.

### **Support RESET**

This selection box permits you to determine whether you want the alarm reset (RESET) activated or not.

### Use Variable TimeStamp

When this property is enabled the alarm ON and OFF times will include a TimeStamp (date and time of last modification executed) of the variable that generated it. The Variable's TimeStamp is set from the communication driver and the OPC management.

In addition to this and as usual the event in question will be recorded in the Historical Log with the TimeStamp of the variable that generated it.

This function could be handy in situations where the variable's value is notified to Movicon with a delay time in respect to its effective variation. For instance, in a OPC communication, the Server may happen to be in Client refresh mode for a few seconds. In this case the variable TimeStamp will nevertheless be received by the Server indicating the actual time in which the variable effectively changed in the Server.

The "Use Variable TimeStamp" can be also read and modified in runtime using the "DateTimeFromTimeStamp" property from the "AlarmThresholdCmdTarget".

### Allow Reset with Condition ON

This property can be disabled to prevent alarms from being reset when their status is still ON.

### Blink

This selection box permits you to determine whether or not to active the alarm blinking function the moment the alarm activates.

### Print

This property is not currently in use. To print alarms you need to use the report functions which can be activated with the "Alarm Commands" from the "Command List".

### Record on Historical Log

This selection box permits you to determine whether or not to activate the recording in the Historical Log of alarms or messages upon the relative ON, OFF, ACK, RESET events. These recordings will be contained in the "Alarms" table of the Historical Log Database created by Movicon in the defined format or in the Alarms.dat file, depending on the data source you are using (ODBC or IMDB)



If you are using the ODBC connection, the Historical Log is created with the Movicon default settings, but the Historical Log file can be customized when put into use, where you can create a personalized ODBC link and define a different table name. These functionalities can be carried out from the Project's "Historical Log Settings".

### Beep

This property allows you to decide whether the alarm threshold should make a beep sound when the alarm is activated.

### Bmp File

This property allows you to set an alarm image. The image will be displayed as an icon in one of the alarm window's columns. The image set in the alarm threshold has priority over the one set in the variable. This property cannot be modified using VBA language.

### Sound File

This property allows you to set a alarm sound wav. file which will activate when the alarm activates. The sound set in the alarm threshold has priority over any one set in the variable. This property cannot be modified using VBA language.

When more than one alarm have been associated an audio file to be executed when the alarm turns ON, this file is executed according to the alarm priorities. For example, when an alarm turns ON, when other alarms associated with audio files, are already ON, the new alarm will executed the audio file only when its priority is greater than those already present. Note that the "1" priority is the lowest.

### Repeat Alarm Sound

This check box permits you to define the behaviour of the sound files associated to the alarms. By activating the property the file will be executed continuously until it is acknowledged. Otherwise the file will only be executed once when the alarm appears.

### Speech Alarm Text

This property allows you to enable the speaking functions for vocal synthesis of the text string which builds the alarm. The alarm's text will then be pronounced by Movicon when it occurs if the PC being used has been equipped with an audio card and speakers. You will

also need to ensure that the phonemes, to be used in pronouncing the text, have been installed with language desired. The phonemes are used to pronounce the text by interpreting the pronunciation of the language selected. The phonemes are normally supplied separately and therefore it is the user's responsibility to get and install the ones desired. However, by using Movicon's customized installation you can select some of the most commonly used phonemes.

#### **Repeat Speech**

This property allows you to set the repeat time of the alarm's Speech. When setting this property with a '0' value the alarm's speech will be repeated once only. This time must also be set allowing enough time for the alarm's speech to be executed.

#### **Enabling Speech Variable**

This selection box is used for selecting a variable which will determine the activation of the alarm's speech. When the selected variable obtains a value different from (><0), the alarm's text will be spoken when the alarm is activated. When the variable obtains a '0' value the Speech will result disabled.

#### **BackColor**

This property is dedicated to selecting the colour to be associated to background relating to the alarms or messages display in the appropriated window.

Selecting colours is done according to the standard selection modes using the colour palette.

You can reset the set colours (restoring those of default) by selecting the 'Automatic' option found on the colour palette.

#### **TextColor**

This property is dedicated to selecting the colours to be associated to the texts relating to the alarms or messages displayed in the appropriated window.

Selecting colours is done according to the standard selection modes using the colour palette.

You can reset the set colours (restoring those of default) by selecting the 'Automatic' option found on the colour palette.

#### **Blink BackColor**

This property is dedicated to selecting the colour to be associated to the background relating to the alarms or messages displayed in the appropriated window during the blinking phase.

Selecting colours is done according to the standard selection modes using the colour palette.

You can reset the set colours (restoring those of default) by selecting the 'Automatic' option found on the colour palette.

#### **Blink TextColor**

This property is dedicated to selecting the colour to be associated to the text relating to the alarms or messages displayed in the appropriated window during the blinking phase.

Selecting colours is done according to the standard selection modes using the colour palette.

You can reset the set colours (restoring those of default) by selecting the 'Automatic' option found on the colour palette.

## **Alarm Threshold Event Notification Properties**

---

The Alarm Threshold Events Notification properties allow you to define the managing of messages on event with the aim of sending Email (by means of the MAPI functions), Vocal Messages, SMS, Fax etc. (by means of the TAPI functions).

To modify the Alarm Threshold Events Notification's settings, select the threshold with the mouse and use the Movicon **'Properties Window'**.

**Note: This is an optional feature: check your dongle options.**



Movicon supports the MAPI (Messaging Application Program Interface) by means of any management tool of the electronic post, whether in Client or Server mode. For example, Windows has been installed with the post Client Outlook Express, which can be exploited for your specific aims. If

you have a post Server (eg. Microsoft Exchange), the operating possibilities can be customized. Installation and configuration of the post tools the modem and access to Internet are to be taken care by the PC's post administrator.



Movicon supports TAPI (Telephonic Application Program Interface) for managing telephone lines. Sending SMS messages, faxes or vocal messages requires the configuring of the appropriated optional functions. In addition to this it will be necessary to configure the relevant tools such as modem or other in function the management being used.

#### **Recipients**

Through this property you can select or type the recipient user or recipient user group to which the message, SMS, E-mail etc., is to be sent.

The user profile, which is defined through the '**Users & User Groups**' settings, must contain a telephone number or E-Mail address or Messenger address in accordance to what is required for sending messages.

#### **Text to Send**

This property allows you to set a custom text which will then be used for SMS, Voice, Email, Fax, Messenger and Admin.Alert notifications. When this property is left without a value, the sent string will be composed in the usual way be Movicon. This property cannot be modified by means of using VBA language.

### **E-Mail**

---

#### **Attachment**

In this selection box you can specify one or more files to be attached to the message to be sent to the recipient when needed. If there are more than one file to be sent use the ";" character as a separator (eg. File1.zip;File2.zip;File3.zip).

If is not specified the absolute files path, Movicon will search for these files within the setting folder for the Alarm Dispatcher (.dspt).

Note that this function is available starting from Build 956.

#### **Send Email ON**

This property activates or deactivates the sending of E-mail messages management. This management requires the E-mail Client and Server management tool installation, as well as the necessary modem and Internet connections.

The message will be sent upon the Alarm's intervention (ON).

#### **Send Email ACK**

This property activated or deactivates the send E-mail message management. This management requires the E-mail Client and Server management tool installation, as well as the necessary modem and Internet connections.

The message will be sent the moment in which the alarm is acknowledged (ACK).

#### **Send Email RESET**

This property activated or deactivates the send E-mail message management. This management requires the E-mail Client and Server management tool installation, as well as the necessary modem and Internet connections.

The message will be sent the moment in which the alarm is reset (RESET).

#### **Send Email OFF**

This property activated or deactivates the send E-mail message management. This management requires the E-mail Client and Server management tool installation, as well as the necessary modem and Internet connections.

The message will be sent the moment the alarm stops ringing, independently from the acknowledge or reset status.

### **Voice**

---

#### **Send Voice ON**

This property activates or deactivates the voice message management. This management requires the enabling and configuring of the appropriate functions for sending voice messages (TAPI), and the installation of any other tools required as well as the configuring of any modem needed.

The message will be sent upon the Alarm's intervention (ON).

**Send Voice ACK**

This property activates or deactivates the send voice messages management. This management requires the enabling and configuring of the appropriate functions for sending voice messages (TAPI), and the installation of any other tools required as well as the configuring of any modem needed.

The message will be sent the moment in which the alarm is acknowledged (ACK).

**Send Voice RESET**

This property activates or deactivates the send voice messages management. This management requires the enabling and configuring of the appropriate functions for sending voice messages (TAPI), and the installation of any other tools required as well as the configuring of any modem needed.

The message will be sent the moment in which the alarm is reset (RESET).

**Send Voice OFF**

This property activates or deactivates the send voice messages management. This management requires the enabling and configuring of the appropriate functions for sending voice messages (TAPI), and the installation of any other tools required as well as the configuring of any modem needed.

The message will be sent the moment the alarm stops ringing, independently from the acknowledge or reset status.

---

**SMS****Send SMS ON**

This property activates or deactivates the send SMS management. This management requires the enabling and configuring of the appropriate functions for sending SMS messages and the eventual modem to be configured.

The message will be sent upon the Alarm's intervention (ON).

**Send SMS ACK**

This property activates or deactivates the send SMS management. This management requires the enabling and configuring of the appropriate functions for sending SMS messages and the eventual modem to be configured.

The message will be sent the moment in which the alarm is acknowledged (ACK).

**Send SMS RESET**

This property activates or deactivates the send SMS management. This management requires the enabling and configuring of the appropriate functions for sending SMS messages and the eventual modem to be configured.

The message will be sent the moment in which the alarm is reset (RESET).

**Send SMS OFF**

This property activates or deactivates the send SMS management. This management requires the enabling and configuring of the appropriate functions for sending SMS messages and the eventual modem to be configured.

The message will be sent the moment the alarm stops ringing, independently from the acknowledge or reset status.

---

**Fax****Send FAX ON**

This property activates or deactivates the send FAX management. This management requires the enabling and configuring of the appropriate functions for sending FAXES and the eventual modem to be configured.

The message will be sent upon the Alarm's intervention (ON).

**Send FAX ACK**

This property activated or deactivates the send FAX management. This management requires the enabling and configuring of the appropriate functions for sending FAXES and the eventual modem to be configured.

The message will be sent the moment in which the alarm is acknowledged (ACK).

**Send FAX RESET**

This property activates or deactivates the send FAX management. This management requires the enabling and configuring of the appropriate functions for sending FAXES and the eventual modem to be configured.

The message will be sent the moment in which the alarm is reset (RESET).

**Send FAX OFF**

This property activates or deactivates the send FAX management. This management requires the enabling and configuring of the appropriate functions for sending FAXES and the eventual modem to be configured.

The message will be sent the moment the alarm stops ringing, independently from the acknowledge or reset status.

---

**Admin.Alert****Send Admin.Alert ON**

This property activates or deactivates the send Admin.Alert messages.

The message will be sent upon the Alarm's intervention (ON).

**Send Admin.Alert ACK**

This property activates or deactivates the send Admin.Alert messages.

The message will be sent the moment in which the alarm is acknowledged (ACK).

**Send Admin.Alert RESET**

This property activates or deactivates the send Admin.Alert messages.

The message will be sent the moment in which the alarm is reset (RESET).

**Send Admin.Alert OFF**

This property activates or deactivates the send Admin.Alert messages.

The message will be sent the moment the alarm stops ringing, independently from the acknowledge or reset status.

---

**Messenger****Send Messenger ON**

This property activates or deactivates the send messages management through the Microsoft MSN Messenger service. This management requires the enabling of the MSN Messenger service and an Internet connection.

The message will be sent upon the Alarm's intervention (ON).

**Send Messenger ACK**

This property activates or deactivates the send messages management through the Microsoft MSN Messenger service. This management requires the enabling of the MSN Messenger service and an Internet connection.

The message will be sent the moment in which the alarm is acknowledged (ACK).

**Send Messenger RESET**

This property activates or deactivates the send messages management through the Microsoft MSN Messenger service. This management requires the enabling of the MSN Messenger service and an Internet connection.

The message will be sent the moment in which the alarm is reset (RESET).

**Send Messenger OFF**

This property activates or deactivates the send messages management through the Microsoft MSN Messenger service. This management requires the enabling of the MSN Messenger service and an Internet connection.

The message will be sent the moment the alarm stops ringing, independently from the acknowledge or reset status.

# Alarm Dispatcher

---

The **Alarm Dispatcher** is a software program for sending message and alarm notifications by using the various configured communication methods (PlugIn). The available communication technologies are:

- **SMS messages using SMPP protocol**
- **SMS messages using GSM**
- **Faxes**
- **Voice Messages (Vocal Synthesis)**
- **E-mails using SMTP protocol**
- **E-mail using MAPI**



The **Alarm Dispatcher** is accessed through the Movicon "Tools Menu". For further information on configuring the **Alarm Dispatcher** please refer to the specific program's Help (Appendix).

The dispatcher can manage message notifications with programmed delays, according to the severity of the alarm ("Settings->General"). All the pending messages, which therefore have not been sent, are rendered persistent by Movicon in the "ADPending.xml" file saved in the project's DATA sub-folder. In cases where Movicon is started up with messages still pending, the Alarm Dispatcher will also automatically start up to allow these pending messages to be sent.



The above described pending message manager is NOT supported on WinCE.





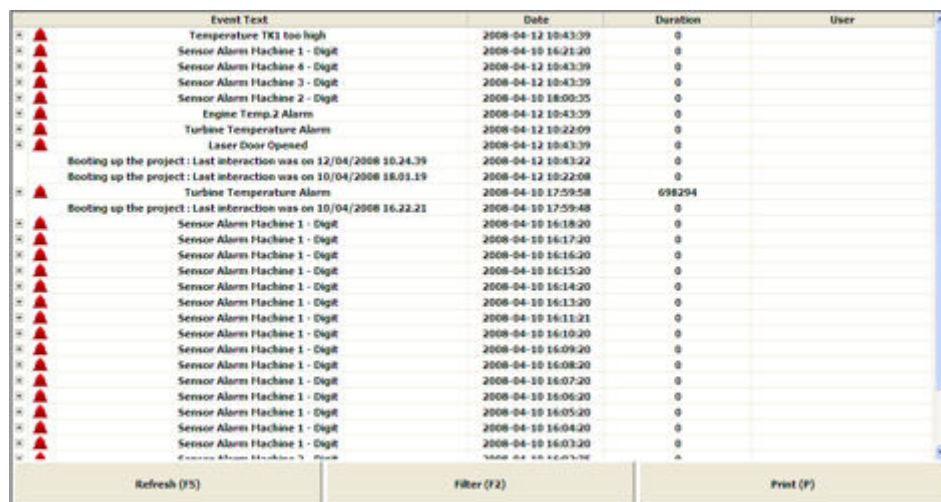
# 19. Historical Log

***A project's Historical Log automatically manages the recording of plant alarms or events on archive files in database or text format.***

In every supervision system the sequence of all significant events (Alarms, Messages) which verify during a project run must be recorded. Event recordings is therefore a basic functionality, and permits historical analysis of everything that happened while the plant was working. The Movicon function which records events on Log files is called the **Historical Log**.

The Historical Log's task is to record chronologically all the project events (Alarms, System Messages, Drivers Messages) which happened during the applied progress runtime process.

Recording takes place, according to the modalities described in the "**Historical Log Management**" chapter, which Movicon writes in appropriate files and displays logged information in the Log Window, one of the graphic objects which can be inserted from the toolbox into project screens.



Event Text	Date	Duration	User
Temperature TK1 too high	2008-04-12 10:43:39	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:21:00	0	
Sensor Alarm Machine 4 - Digit	2008-04-12 10:43:39	0	
Sensor Alarm Machine 3 - Digit	2008-04-12 10:43:39	0	
Sensor Alarm Machine 2 - Digit	2008-04-10 18:00:35	0	
Engine Temp.2 Alarm	2008-04-12 10:43:39	0	
Turbine Temperature Alarms	2008-04-12 10:22:09	0	
Laser Door Opened	2008-04-12 10:43:39	0	
Booting up the project : Last interaction was on 12/04/2008 10:24:39	2008-04-12 10:43:22	0	
Booting up the project : Last interaction was on 10/04/2008 18:01:19	2008-04-12 10:22:08	0	
Turbine Temperature Alarms	2008-04-10 17:59:58	698294	
Booting up the project : Last interaction was on 10/04/2008 16:22:21	2008-04-10 17:59:48	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:18:20	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:17:20	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:16:20	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:15:20	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:14:20	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:13:20	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:11:21	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:10:20	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:09:20	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:08:20	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:07:20	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:06:20	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:05:20	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:04:20	0	
Sensor Alarm Machine 1 - Digit	2008-04-10 16:03:20	0	

The events which can be recorded in the Historical Log archive are:

- System operations (operator moves on system)
- System diagnostics (auto-diagnosis, Driver diagnosis)
- All the project Alarms considered important by the programmer
- Log On and Log Off of users
- Status change of variables associated to the Trace function
- The contents of plant variables associated to project strings on event established by the programmer
- Basic script logic messages coming from the "Debug.Print" function
- Basic script logic messages coming from the Trace functions
- Messages deriving from other applications established by the programmer

All the project alarms, if not specified otherwise in the properties, are recorded in the Historical Log. The programmer can then configure the alarm properties not to record in the historical log if desired.

The System and Driver Events are always recorded in the appropriate Log files. If you don't wish them to be displayed, set the log data display in the log window appropriately.

The recording of events and the database format must be configured in the **"Project Historical Log Settings"**.

The log data can be displayed through the appropriated **"Log Window"** and **"TraceDB Window"** windows which can be inserted into any project screen.

## Log File Management

---

Movicon provides the possibility to use two different types of historical logs in managing files on disk:

- Standard ODBC
- IMDB

When using the standard ODBC, Movicon will use relational database tables and files as log files. Normally, files in this case are MsSQL Server or MsAccess type, but the ODBC formats of any Relational DB can be used (ie. Oracle or other).

The two different technologies are described in the corresponding chapters.

The type of archive choice does not change the fundamental way the Historical Log is used. In both cases Movicon will record events on files, and their contents will be automatically displayed in log windows.

- Nevertheless, it would be handy to remember that the use of standard ODBC offers you the advantage of managing a database in Relational DB, with many potentialities in terms of analysis and statistics with the use of the SQL language. However, in some cases the ODBC technology may be too much in terms of performances and resources required in reduced hardware devices such as embedded systems or those based on Windows CE (HMI or Mobile).

In conclusion the historical log format choice is, of course, in the hands of the programmer. Movicon, for default, will use the ODBC in desktop systems based on Win32 and IMDB in embedded systems based on Windows CE. The programmer may decide whether to use IMDB on desktop or force the use of ODBC in WinCE which requires ADOCE pre-installation to convert the ODBC links automatically in ADOCE in the device. (ODBC is not supported by Window CE).



Note that with creating projects for Windows XP/Vista, Movicon permits activation of automatic ODBC link creation to historical logs. IMDB use is not selected for default. When nothing has been selected, Movicon will use the ODBC links with historical logs based on SQL Server where possible, otherwise it will use MsAccess.

When creating projects for WinCE, Movicon will select the historicals based on IMDB for default.

All the default settings can be modified as pleased.

### Log File Size

---

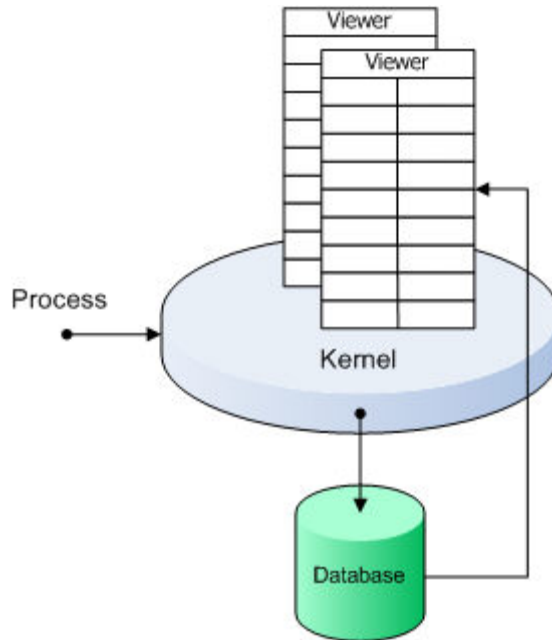
The size of the log files is set in the **"Project Historical Log Settings"** item in the project settings. The size or age is expressed in days and has been set at 180 days for default. This value can be set as desired.

The file size therefore depend on the data format used and by the number of events to be recorded by the system in the reset period.

## Database Log File

The Database file containing Log information will be created by Movicon automatically in the project's 'LOGS' folder. Based on the file type chosen, the data available will be as indicated below.

In any case, the Log display windows access the files to show logged events on screen, independently from the type of file being used.



## ODBC

Normally, in the Win32 environment, the use of logs based on Relational DB offers many advantages in terms of managing files, openness, analysis and statistics. Movicon, if not specified otherwise, will use the ODBC links for managing Log files on Win32 systems.

The Log file names, if not specified otherwise, will be "ProjectName\_hisLog" in the project's "LOGS" folder. However, you can customize the file name and the **ODBC** links through the "**Project Historical Log Settings**" properties. Three different tables will then be created in the Database, each one will contain the data relating to a certain type of event.



Movicon, if not specified otherwise, will use ODBC in Win32 environments. The format it will use is MsSQL Server or MsAccess as an alternative. When unable to create files and connections automatically, Movicon will alert with a message where you may then need to use the ODBC link manually or use the IMDB.

## IMDB

The log file management in **IMDB** (InMemory DB) is a valid alternative to Relational DB, which is ideal for embedded systems with limited resources or based on Windows CE or Pocket PC.

The IMDB does not use ODBC and managed log files in text mode, by managing data in memory and unloading it on files at preset time interval. Filed data can be based on XML, on simple text or crypted.

The advantages of using this management as an alternative to the potentialities used by the ODBC are:

1. Increased performances
2. Minimum resources required
3. Date base can be crypted against unauthorized access.



Movicon, if not specified differently, will use IMDB in WinCE environments. When the programmer is obliged to setup the ODBC management, Movicon will automatically convert the project's ODBC links to ADOCE links in the device. This however, requires the implementation of the ADOCE components on the device.

## Data Tables

The Log file archive tables are:

- **Alarms:** contains the project's alarm messages
- **Drivers:** contains messages coming from the project's Communication Drivers
- **SysMsgs:** contains the project's system messages

Each one of these tables has the following structure:

- **Time Column (TimeCol):** reports the date and time of the recording in GMT (Greenwich Mean Time)
- **Local Time Column (LocalCol):** reports the date and time of the recording in local time
- **MSec Column (MSecCol):** reports the milliseconds relating to the time of the recording
- **User Column (UserCol):** reports the name of the user who was active at the time the recording took place
- **Event Column (EventCol):** reports the event type recorded (Eg. Alarm ON, Alarm OFF, System, ecc)
- **Event Number Column (EvNumCol):** reports an ID number of the event recorded
- **Event Description Column (EvDescCol):** reports the event description
- **Description Column (DescCol):** reports information about the event type
- **Comment Column (CommCol):** reports information about the event type
- **Duration Column (DurCol):** reports the duration time of the event in question
- **Unique ID Column (UniID):** reports the alarm's unique ID (value valid for the "Alarms" table only)
- **Transaction ID Column (TraID):** reports the alarm's transaction ID (value valid for the "Alarms" table only)

However the names of the database columns can be customized through the **"Project Historical Log Settings"**.

Event Text	Event Time	Event Id	User
Booting up : Last interact...	2004-09-20 17:...	System	
Booting up : Last interact...	2004-09-20 17:...	System	
Booting up : Last interact...	2004-09-20 16:...	System	
Booting up : Last interact...	2004-09-20 16:...	System	
Booting up : Last interact...	2004-09-20 16:...	System	

Event Text	Event Time	Event Id	User
Alarm Active - Digital : VAR00001	2004-09-20 17:0...	ALARM OFF	
Alarm Active - Digital : VAR00001	2004-09-20 17:0...	ALARM ON	

Event Text	Event Time	Event Id	User
Communication error : stati...	2004-09-20 17:0...	Com. Dr...	
Communication error : stati...	2004-09-20 17:0...	Com. Dr...	

An Example of a Log Window containing data taken from the three log tables.

## 20. Historical Log Management

***The Movicon recording engines allow process data to historically logged in two alternative modes. You can either use the ODBC standard or the IMDB technology.***

Usually, a supervision project has, among other tasks, to record and log process data on files, whether this be significant events or messages, process information (variables/tags) which are traced and recorded at timed intervals on event or change.

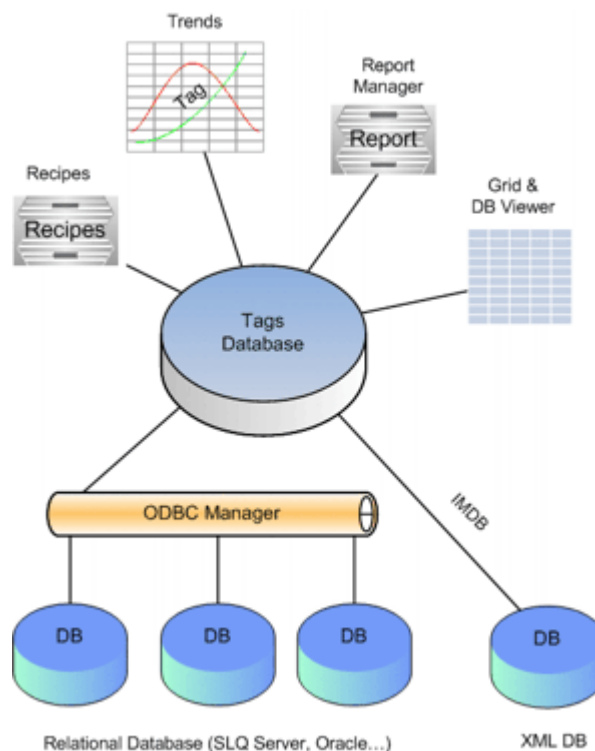
Movicon provides you with the possibility to use two different types of management for recording historical data on files:

- Standard ODBC (Open Database Connectivity)
- IMDB (InMemory DB)

When using the standard ODBC, Movicon will use relational database tables and files as log files. Normally, the files in this case are MsSQL Server type or MsAccess, but ODBC formats from any Relational DB can be used (ie. Oracle or other).

When using the IMDB, Movicon will record historical log data in text mode, which can be structured in XML or crypt.

These two different technologies are described in the corresponding chapters.



A system structure for the historical log management.

The type of archive choice does not change the fundamental way the Historical Log is used. In both cases Movicon will record events on files, and their contents will be automatically displayed in log windows.

- Nevertheless, it would be handy to remember that the use of standard ODBC offers you the advantage of managing a database in Relational DB, with many potentialities in terms of analysis and statistics with the use of the SQL language. However, in some cases the ODBC technology may be too much in terms of performances and resources required in reduced hardware devices such as embedded systems or those based on Windows CE (HMI or Mobile).

In conclusion the historical log format choice is, of course, in the hands of the programmer. Movicon, for default, will use the ODBC in desktop systems based on Win32 and IMDB in embedded systems based on Windows CE. The programmer may decide whether to use IMDB on desktop or force the use of ODBC in WinCE which requires ADOCE pre-installation to convert the ODBC links automatically in ADOCE in the device (ODBC is not supported by Windows CE).



Note that with creating projects for Windows XP/Vista, Movicon permits activation of automatic ODBC link creation to historical logs. IMDB use is not selected for default. When nothing has been selected, Movicon will use the ODBC links with historical logs based on SQL Server where possible, otherwise it will use MsAccess. When creating projects for WinCE, Movicon will select the historicals based on IMDB for default. All the default settings can be modified as pleased.



**CAUTION:** Data Loggers based on IMDB do not support the Reports from Crystal Reports otherwise use the historical logs in ODBC.

## Log File Size

The file size is set in the **"Project Historical Log Settings"** item in the project settings for the Historical Log and is set in the respective Data Logger Recording Time Properties or the DB trace objects. The size or age is expressed in days and has been set at 180 days for default. This value can be set as desired.

The file size therefore depends on the data format used and by the number of events to be recorded by the system in the reset period.

## ODBC Standard

***Movicon fully supports the Open Data Base Connectivity standard allowing data to be saved in standard format and real-time links to external databases.***

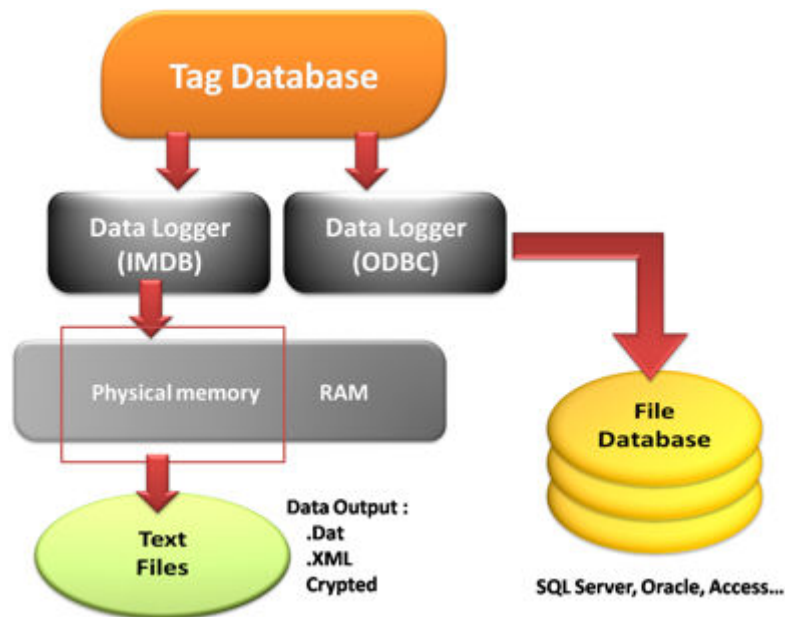
The Open Data Base Connectivity (ODBC) is the Microsoft standard which consents applications to organize data in database files in the specific format of any other application enabled to support this standard.

By using this technique you can knock down the data exchange barriers between different formats.



For example with the ODBC you can record data from the Historical Log or record with the Data Logger and Recipe tables data containing Movicon variables.

The data can be archived, by using the ODBC link, in the user's preferred format. If the user has a management system based on MsAccess™ or SQL Server, s/he can read and manipulate the data recorded by Movicon according to their own requirements.



Movicon carries out data recordings by using the ODBC standard. The ODBC link interprets the data to be recorded according to how it has been configured and carries out the recording in the corresponding format through the ODBC driver of the preferred database application. The file or files will therefore contain Movicon data, which will actually be recorded in the format requested by the ODBC system.



Movicon is independent from the data format used, seeing as the driver is property of the database and the operating system's ODBC manager being the means of putting the writing of data into effect. ODBC is a Microsoft standard and the ODBC manager files (drivers) are property of the respective owner of each single Database application. For further information please refer to the ODBC online guide or to the current bibliography available from Microsoft or the makers of the database being used.

## ODBC Drivers

In order for Movicon to log historical data on files in standard ODBC format, the Windows OS needs to be equipped with the appropriate ODBC drivers for the chosen database product.

During a normal installation Movicon sees to it that the necessary files are installed for the ODBC management and drivers for MsAccess™, Excel™

To install the SQL Server™ drivers you need to execute a 'Customized' installation and select the MSDE item (Microsoft Data Engine). In this way Movicon will install the MSDE kit which integrates only the RunTime part of the SQL Server database management.



Movicon is preset to install the ODBC drivers of the Microsoft Access, Excel and SQL Server MSDE products.

When using other database products you will need to check whether the relevant ODBC driver supplied by the manufactures has been installed.

The operating system, however, provides a series of ODBC drivers already available in different formats (Paradox, FoxPro, dBase, ecc.), which can be accessed just like all the other ODBC settings from the '**Administrator Tools - ODBC Data Source**' item on the '**Control Panel**'.



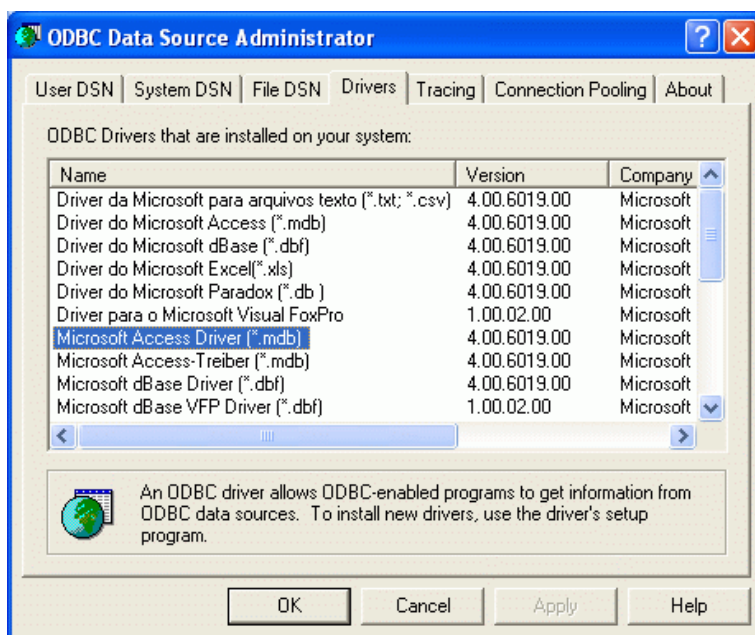
## ODBC Links

In order for an application to record data by means of the ODBC standard, as well as to dispose the chosen data format of the ODBC driver, you will need to create a **'Link'** between the Server application, the table and the destination file, in which the values from the ODBC driver will be written.

It is through this 'Link with the ODBC driver that enables a resource from a Movicon project application to record data on file. The ODBC links must all be registered in the operating system through the appropriate ODBC system settings from the **'Administrator Tools - ODBC Data Source'** item on the **'Control Panel'**. However, Movicon is provided with a guide tool (Auto-configure) to create ODBC links for all the resources or functionalities permitting their use.



The ODBC link is the tool through which an application is linked to the ODBC driver so that it can write or read data using an external format. The ODBC driver standard is based on the specifications defined by Microsoft.



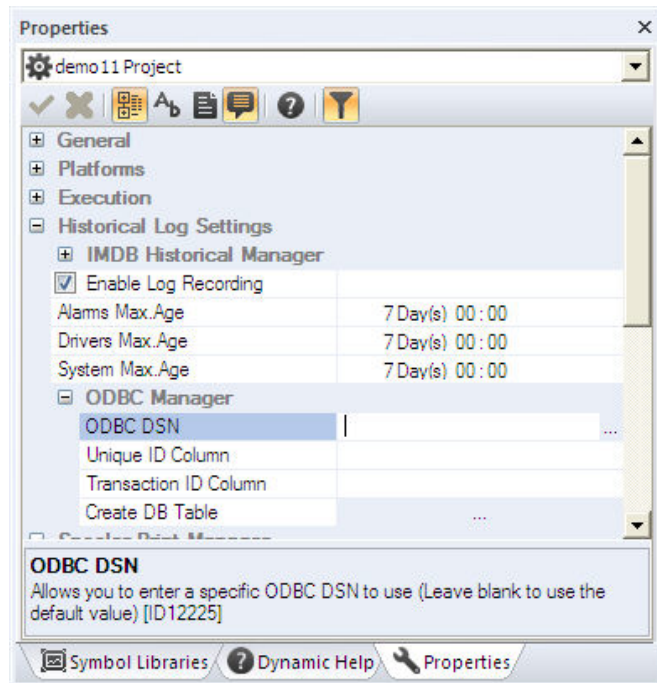
This window shows the list of ODBC drivers usually currently existing in the Windows OS, according to the installation standards. Other drivers can be added when installing other compatible ODBC applications.

ODBC link can be configured by setting the data manually, by activating the ODBC system from Movicon or from the Windows 'Control Panel', by activating the icon as shown in the above figure.

To make the configuring operation easier, Movicon automatically creates the ODBC links and the database files relating to the most commonly used components being Data Loggers, Recipes, Historical Log, Variable Tracer, etc.

These links and files created for default by Movicon naturally have a predefined name which also includes the project's name. The automatic creation consents you to create links (eg, file and/or table if required) in MsAccess™ format with only one command.

However, you can execute the create a customized ODBC link and relating database file by exploiting a **Wizard** left at your disposition by Movicon to make this operation easier, so you don't have to enter the configuration through the Operating System's 'Control Panel'. The resources which can interact with the database through ODBC links have a field in their 'Properties Windows' for creating customized ODBC links. The Wizard procedure, for creating the links, is started by clicking the "..." button positioned on the right hand side of the edit box:



If the user is using a SQL Server™, Access™ or Excel™ database format, Movicon will remarkably simplify the ODBC links management by automatically creating the necessary manoeuvres with just one command.



ATTENTION! Even though Microsoft also provides a ODBC driver for Excel™ it is not a Database, but an electronic sheet. It is for this reason that Excel™ does not support all the SQL commands compatible with the most common Databases and therefore Excel™ cannot be used as if it were a database. The Movicon resources, such as the Data Loggers, Recipes, Historical Log, Variable Tracer, etc., cannot therefore be managed through ODBC links to Excel™ files.

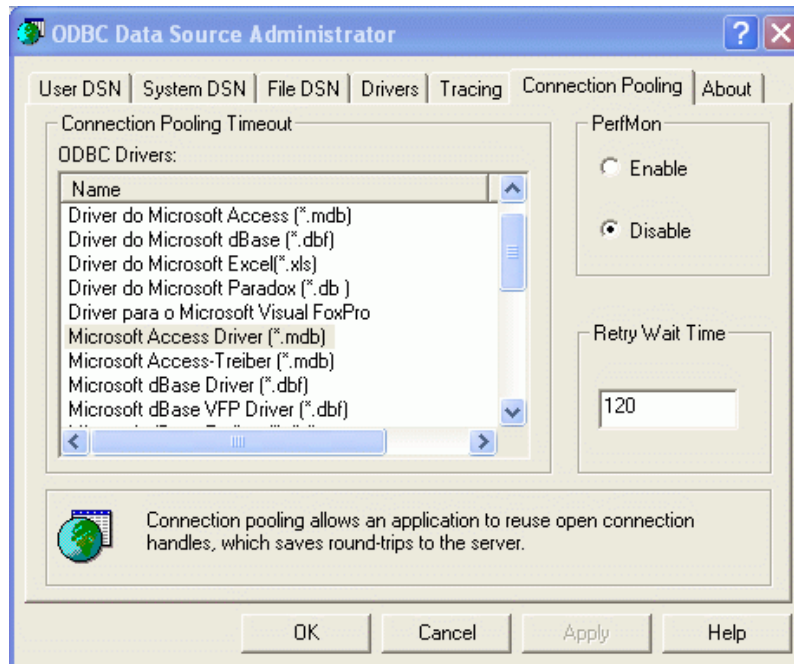


**The ORACLE's ODBC driver has some special requirements that force the programmer to observe some specific rules creating the project links. These rules are:**

- a. All the column's names **must be written with upper case**. Due to this, it's necessary to change the standard column name of any historical table, using the related properties. For instance, the column TimeCol must become TIMECOL.
- b. Spaces inside columns and tables name are not allowed.
- c. Columns and Tables names cannot begin with a number.
- d. Some data types are not supported, and the Oracle ODBC driver uses NVARCHAR also for numeric types (byte, word, integer). This is not a problem when data are displayed using the standard Movicon object windows.

## Connection Pooling

Starting from the 3.5 version of the Windows ODBC manager a '**Connection Pooling**' tab has been added in which you can enable the pool management for each single ODBC driver.



The connection pooling consents the user's application to use one of the available connections in a pool so it does not have to be restored each time it is used. This is because the moment a connection has been created, it is placed in a pool to be re-used by the application to avoid having to carry out the whole connection procedure again hence improved performances.



Always check whether the ODBC driver supports the pool management before enabling it.

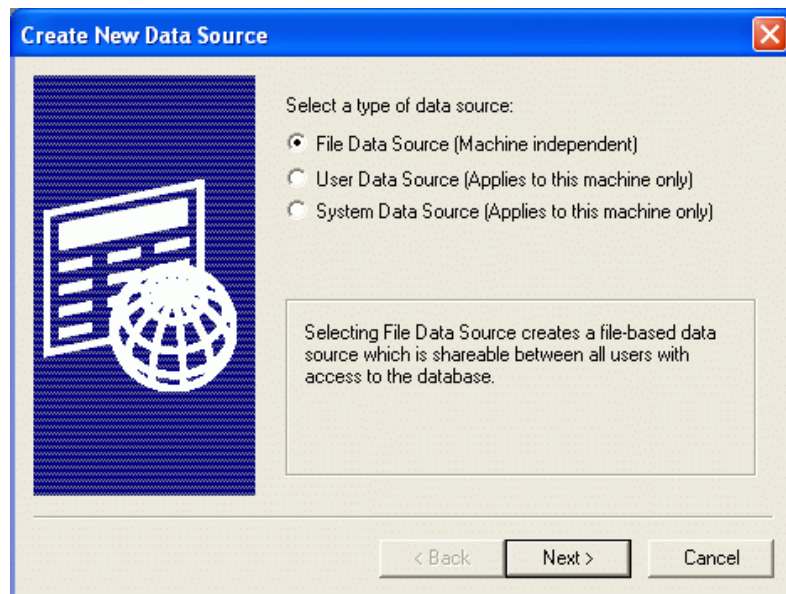
## Creating ODBC links automatically

The first thing Movicon does at Startup is check for any existing ODBC links and only when none exist does it create one by using the default provider (changeable with the registry key "UseDefaultSQLServer"). An exception is made for the 'Recreate All' command in the properties of the resources which use ODBC. In this case the link is created (using the default provider), when the existing one does not allow database connectivity.

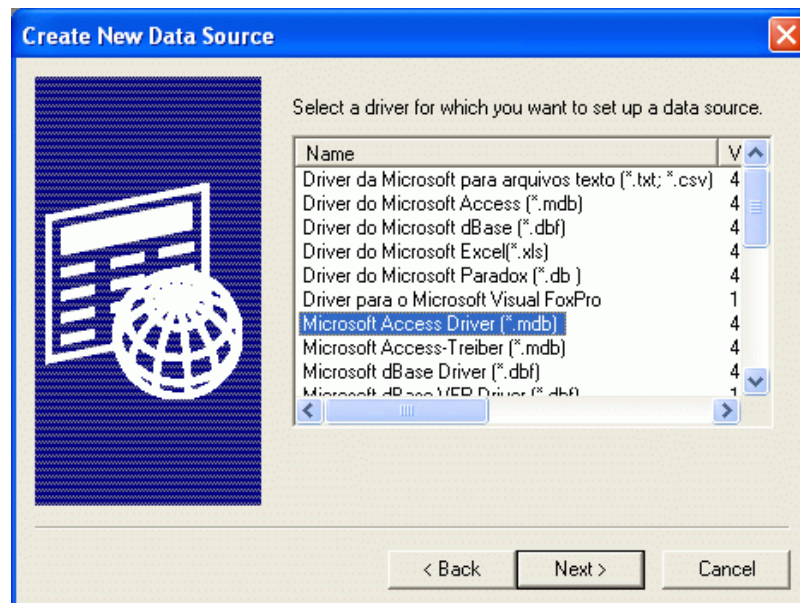
## Creating ODBC Links Manually

By selecting the customized create ODBC Manual Link, you will need to configure the necessary data to get an ODBC link. The procedure, as already seen, is a Movicon Wizard, therefore the programmer will be guided through each step. When clicking on the "..." button found on the right hand side of the "**ODBC DSN**" property introduction field of the resources which support the ODBC links, the following procedure will display:

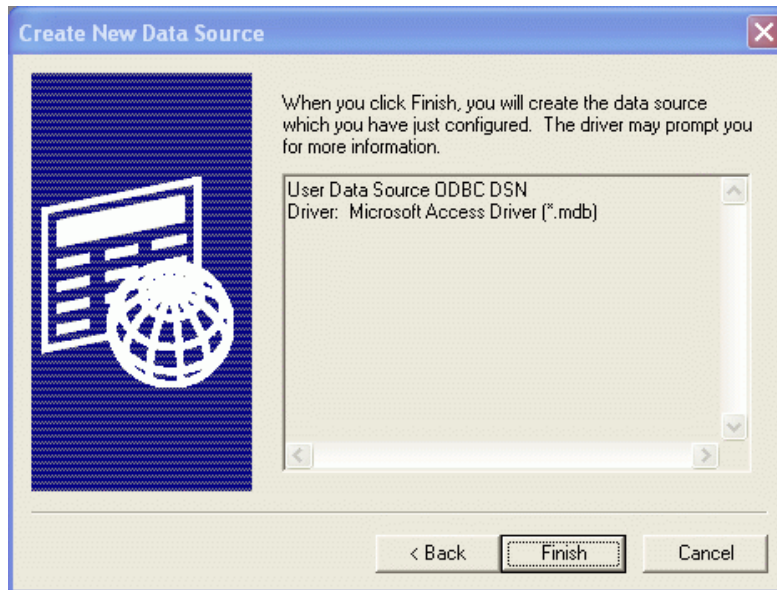
1. The first thing you will be asked is to select the type of ODBC data source. The most common selections are usually 'User Data Source' or 'System Data Source':



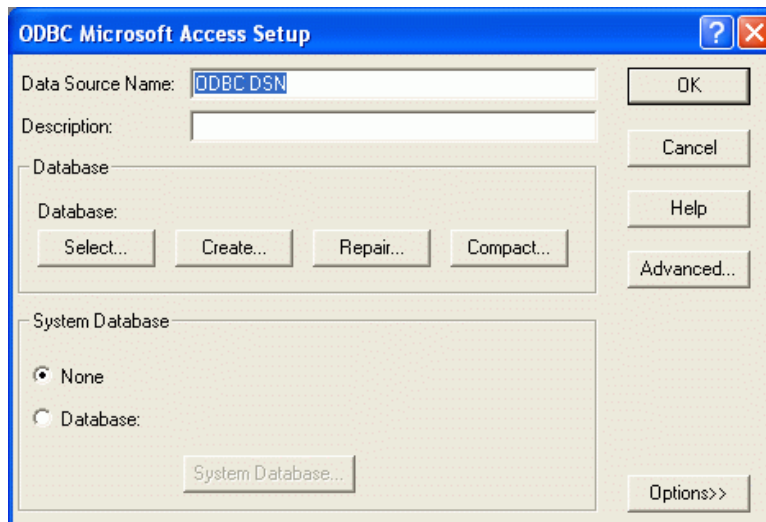
2. At this point you will need to select the ODBC driver you intend to use:



3. **From this moment onwards the configuration windows which will be proposed depend upon the type of driver you selected. Therefore we suggest you consult either the application's manual or the Microsoft ODBC guide or the Database application to be linked.** To continue our demonstration we will select the Microsoft Access driver as an example, seeing that it is the most used, to display the following window:



4. The last proposed window will permit you to define the name of the ODBC link (data source name) and the associated Database file:



If the Database file already exists in the PC, you only need to use the **'Select...'** button to select it. On the other-hand if the file has not yet been created you will have to create it by using the **'Create...'** button.  
Confirm and terminate the create ODBC Link procedure with the 'OK' button.

## SQL Server with MSDE

If you decide to use the SQL Server format, Movicon permits the Microsoft MSDE, the SQL Server standard, to be installed. This SQL product requires the user to refer to the instructions reported in the SQL Server's user's manual.

The Microsoft MSDE service is installed from the Movicon installation CDROM by executing a 'Custom' installation and selecting the MSDE (Microsoft Data Engine) item.

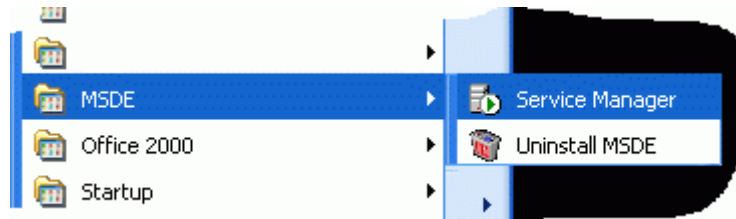
The Microsoft MSDE Manager provides the use of the SQL Server ".MDF" formats by managing tables customized within the SQL Server Database.

If you need to re-install the MSDE service, it can be done by launching the 'MSDEx86.exe' file from the Movicon CD ROM directory with the same name by following the installation

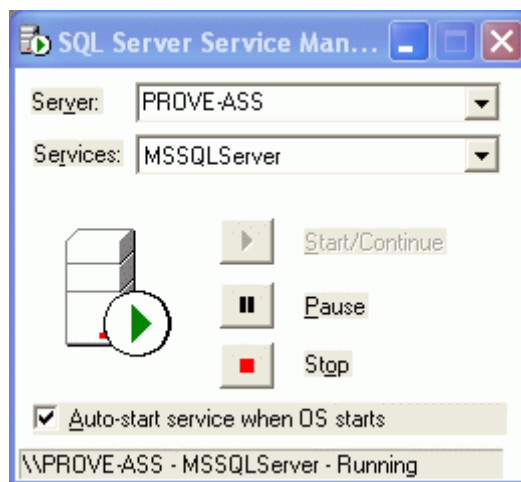
information reported in the 'readme.txt' file. If any errors are signaled during the installation phase you will have to refer to the 'Setup.log' file and the more specific 'Sqlstp.log' file which are found in the Windows installation directory.

The Service Microsoft MSDE service has to be running before any ODBC links with the SQL Server data format can be carried out. In order to do this carry out the following procedure:

1. Install the Microsoft MSDE service from the Movicon installation CD, if not already done so
2. Run the MSDE Service Manager from the Windows start menu, if not already active



3. Put the MSDE engine in Start/Continue, if not already done so:



4. Check whether the MSDE icon shows in the Windows application bar:



5. From this point onwards you can go ahead with creating the ODBC links in Microsoft SQL Server format.



**The tables created in MS SQL Server are created with a "clustered" index in the "TimeCol" column. this allows the highest performances to be obtained when retrieving data sorted out according to that column, being operations which Movicon objects usually do.**

## Using the ORACLE

---

Movicon consents to recording data in ORACLE databases requiring that the ORACLE ODBC driver be installed.

The ORACLE ODBC driver however has certain characteristics that obliges the user to observe a few regulations when creating projects. These regulations are:

1. All the column names must be written with capital letters (uppercase). Therefore this requires that the stand column names of the project's historicals be changed through their appropriate properties. For instance, TimeCol would become TIMECOL (or with a custom name but in capitals)
2. Spaces are not allowed in the table and column names
3. Table and column names cannot begin with a number
4. Many data types are not supported, and the Oracle ODBC driver also uses NVARCHAR for numeric types (byte, word, integer). However, this will not cause any problems when data is displayed in the standard Movicon windows

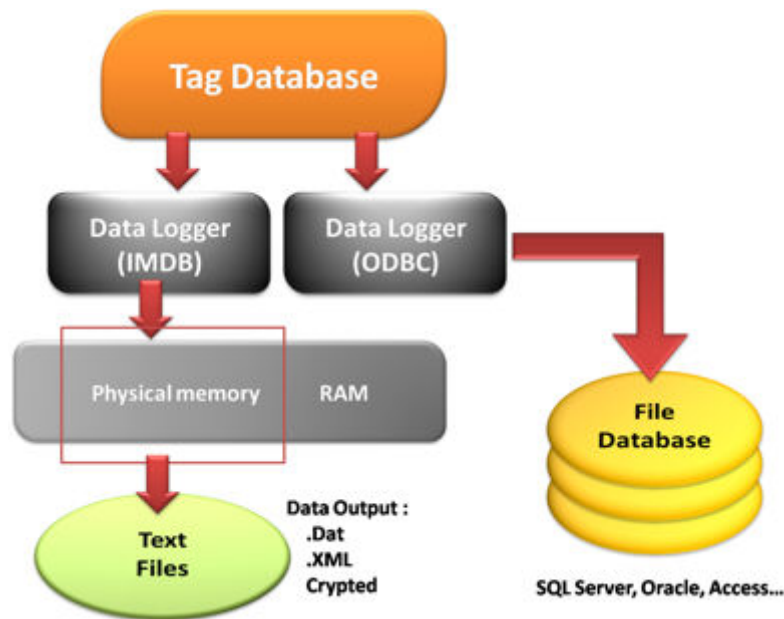
Creating ODBC connections to the ORACLE database in Movicon projects must be done manually by inserting a custom DSN in the historical Log "ODBC DSN" properties. In addition to this, a user must be created with the same name and password of a user defined in the ORACLE database. This Movicon user must then be inserted in the historical log "ODBC DSN User" property.



# IMDB (In Memory DB)

***The IMDB manager (In Memory DB) of historicals allows data to be saved in text and xml format.***

The Movicon historicals management can be executed through the IMDB manager which allows data to be saved in text and XML format. This functionality allows you to get an interesting alternative to the ODBC Manager, allowing you to get log files in text mode by using the XML format structure. By doing this you get the advantage of getting better performances by avoiding the 'layer' of ODBC software and the possibility to crypt data in propriotor format, even if you give up using relational Databases formats.



By using the IMDB in devices based on Windows CE, there is no longer the need to convert the ODBC links to ADOCE, thus the ADOCE component and the SQL Server CE installed on the device are no longer needed which will improve the device performances.

The IMDB manager, with the exclusive Progea technology, manages a memory buffer in which data is placed and then unloaded on file at fixed time intervals. The unloading data on file time can be configured through the appropriate properties of the log engine (ie. Data Logger, Log).

All the data query functions are supported as if managed by a database manager. This function can be activated with a specific option in the historical log, data logger and in the db trace. It is not possible to use this technology in the "Real Time DBMS" as it wouldn't be necessary to use it in that context.

Data is recorded on text files with the ".dat" extension. When the project is started up, Movicon will create table in memory by loading data from these back-u' files. The files can also be saved in XML format.



The "UNION" clause for the data extraction query is not supported by the IMDB. Therefore the "Historical Log Window" will only display system messages even when set to view all message types.



CAUTION: Data Loggers based on IMDB do not support Reports from Crystal Reports otherwise you need to use log file on the ODBC.



## IMDB Settings for Recording data

---

Some of the parameters which are used for the IMDB parameters are the same ones used whether recording a historical Log, a Data Logger or a variable trace, etc. The significance of these parameters are described Below:

### Use IMDB Manager

Enables the use of the IMDB manager to record data. In this case data will not be recorded in DataBase format through the ODBC drivers.



The "UNION" clause for the data extraction query is not supported by the IMDB. Therefore the "Historical Log Window" will only display system messages even when set to view all message types.

### Shared Tables

When this property is enabled, the system's global memory will be used for saving data and not that of the process. This option is especially indicated when using WinCE as operating system because the Movicon CE process memory has a limit of 32 Mbytes. Further more this option is needed also when wanting to access data through external ActiveXs or libraries. In this way the IMDB tables are made available in ram and can be accessed using appropriated ActiveX or library functions.

### Save XML File

Data is always saved in text files with the ".dat" extension and by enabling this option will also allow data to be saved in files in XML format.

### Crypt File

The ".dat" file used for recording data is crypted with a Progea proprietary algorithm and with the public domain algorithm. XML files are not crypted.

### Write Behind Delay

This setting represents the time, in seconds, when the data is unloaded from memory to files.

### Max Nr. Records

This setting allows you to define the maximum number of records the Historical Log can execute. Once this limit had been reached, the data will start recycling. However, data with ages older than the maximum age set will be deleted even when the max. number of records has not yet been reached. This function is disabled when set at the '0' value which means that there are no restrictions to the number of records that can be executed.

## IMDB Database Limitations

---

When using the IMDB manager for recording historical data you will need to keep in mind that there are some limitations that a more conventional database would not have using ODBC drivers. Below you will find a list of the main SQL commands supported by IMDB and its restrictions:

### SQL Language

---

The IMDB manager supports a significant subset of SQL database language functions. The following instructions are particularly supported:

SELECT, UPDATE, INSERT, DELETE  
CREATE TABLE...[PRIMARY KEY...]  
DROP [TABLE]  
SELECT qualifiers: DISTINCT, TOP n  
SELECT clauses FROM, WHERE, GROUP BY, HAVING, ORDER BY  
WHERE expressions: AND, OR, NOT, LIKE, BETWEEN, + - \* /, IS [NOT]  
NULL, <, >, =, < >, <=, >=, Constants, Parameters, ColumnNames  
SELECT list expressions: MAX, MIN, AVG, SUM, COUNT, +, -, \*, /,  
Constants, Parameters, ColumnNames  
Value list qualifiers: ANY, ALL, SOME, IN  
UPDATE expressions: +, -, \*, /, Constants, Parameters, ColumnNames

INSERT values expressions: Constants, Parameters  
INSERT ... SELECT  
Subqueries within SELECT statements

### Limitations

The following limitations must also be taken into consideration:

- Table and Column names cannot contain spaces or any punctuation characters and must start with an alpha character (names cannot start with numbers)
- Views are not supported
- Select list field aliases names are not supported
- Security statements such as COMMIT, GRANT and LOCK are not supported
- The LIMIT TO nn ROWS clause used to limit the number of rows returned by a query is not supported. Use TOP instead to return the top nn rows of a query
- Outer Joins are not supported
- No indexes create and drop statements. Indexes must be created when creating tables

### Data Types

---

The IMDB manager supports the following types of data for table columns:

Numeric	4/8 bytes (32/64 bit value) INT, INTEGER, SHORT, LONG, SMALLINT
Decimal	8 bytes (DOUBLE) REAL, FLOAT, DOUBLE, SINGLE, CURRENCY
Bool	1 bit BOOL, BOOLEAN, YES/NO
Counter	8 bytes Auto-Increment value COUNTER
Character	1 byte per character (2 bytes for UNICODE) Zero to a maximum of 256 MB/row CHAR(n), VARCHAR(n), CHARACTER(n), TEXT
Date/Time	8 bytes 0 to year 20.000, Time 00:00:00.000.000.0 to 24:59:59.999.999 100 nano-seconds resolution DATE, TIME, DATETIME
Binary	Zero to a maximum of 256 MB/row BLOB, LONGBINARY
GUID	For storing Guid values

### Database Default Limits

---

The IMDB manager has the following default limits:

Maximum data store size	~ 2 GB
Maximum number of tables	65535
Maximum number of columns on table	256
Maximum number of indexes on table	64
Maximum length of table names	64
Maximum length of column names	64
Maximum number of rows in a table	2 Billion (2*10^9)
Maximum length of fixed-length column	4096
Maximum length of variable-length column	256 MB
Maximum size for binary columns	256 MB

Maximum number of cursors opened simultaneously	Unlimited (memory depending)
Maximum number of columns in an index	1
Maximum number of XML nodes in a table	2 Billion ( $2 \times 10^9$ )
Maximum number of simultaneously active IMDB objects	Unlimited (memory depending)

## DBMS Settings for Recording data on DataBase

Some parameters which are used for DBMS links are always the same ones used, independently whether recording a Historical Log, a Data Logger, variable tracing, etc. These parameters are described below:

### Keep the DB Connection open

This property, when enabled, allows the **ODBC** connection to be kept open and used for all the transactions to be executed. When the property is disabled, the **ODBC** connection will be opened when a transaction is requested and then closed again.



We suggest you disable the 'Recycle DB Connection' property only when recording is less frequent.

### Send Admin.Alerts

When this property is enabled, each time an error is generated in the current **ODBC** connections the operating system will open an error alert message window by means of Windows' 'Messaging Services'. The "**Enable Admin.Alerts**" property must also be enabled in the 'Project Run Settings'. The error alert message can also be sent locally to the PC where the Movicon project is being run or to a PC connected in net. This is to be specified in the "**Server Admin Alerts**" property in the 'Project Executing Settings'.

### Max. Error Number

The highest number of DBMS errors allowed after which the connection is considered not valid and data is saved on files in ASCII format ("DLOGGERS", "LOGS", "DATA").

### Max. Transitions

Maximum number of transitions per cycle to be updated before closing.

### Max. Cache Size

This setting allows you to set the Cache's maximum size before they system loads data on file. The number is set in Bytes.

### Max. VarChar Precision

This setting allows the maximum precision to be set for the string type columns. The set number represents the number of string characters.

### ODBC DSN

This setting allows you to set a customized **ODBC** connection. Movicon will create a file in Access2000 format in the project's 'LOGS' folder for default with the name:

*ProjectName\_HisLog.mdb*

However you can also customize the **ODBC** connection through this property, creating a different data base from that of Access2000 with a different name.

Unfortunately the ODBC Manager does not pass on any information about the name of the any created connections. Therefore the procedure requires that the name of the connection be inserted first so that Movicon can pass it over to the ODBC Manager. The name of the connection can also be inserted after it has been created. In any case the name of the connection has to be typed in the box manually.

### ODBC DSN User

This property specifies the users name to be used for the **ODBC** connection.

# 21. Users and Passwords Management

---

*Using the Users and Passwords management in projects allows access to protected functions be managed. The project Users can also be used for sending SMS, Voice and Fax messages.*

When using supervision projects it is often necessary to protect the managing of the plant's conditions against unauthorized persons by granting access to the system's functions only after the user's authentication has been verified and confirmed.

Plants can be protected by using the Users and Passwords management.

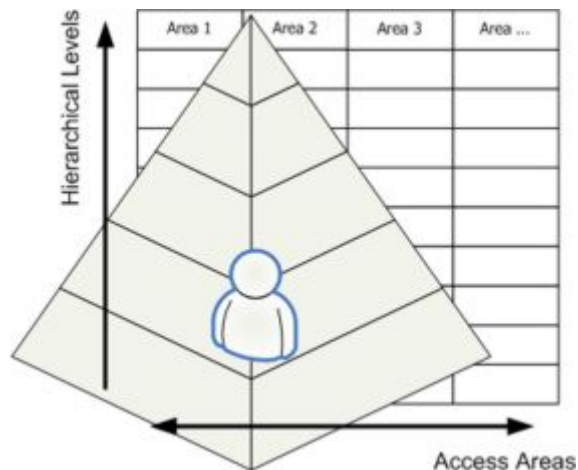
The Users and Passwords management is a supervision project to allow the possibility to manage supervisor access security, by declaring identification of users enabled to work in the system, by entrusting them with an exclusive password and access level. The access privileges (hierarchical levels and Areas) can be issued normally or by group.

- **there are 1024 hierarchical levels and 16 area levels**
- **Log On will not be requested when the objects have been left with the default value (level 0)**

The project must request Log On for each command and function to be protected from unauthorized access. **The hierarchical level must be declared** in the properties of "sensitive" objects to be able to execute the command and the **area it belongs to**.

The execution of these objects will therefore be subject to a Log On to verify the access privileges of the user executing the command.

In this way, any supervisor function can be protected with this type of user level and user password settings verification, according to the modalities and settings described in this chapter.



Movicon allows complete password management, with a unlimited number of users and user groups each one can be set with a level (1-1024) and area (1-16).

# User Levels and Access Levels

---

Each user is assigned a User Password Level for user authentication and a Access Level, which determines which read and write rights the user has on the various project controls, in the Movicon user management.

## User Password Levels

---

The User Password Levels are needed for user authentication. When a user has logged on their Password level is activated enabling the user to execute all the commands protected by a password levels the same as or lower than theirs. The User Password Levels are:

- **User Level 0 (0)** = no password level required
- **User Levels (1-1022)** = password levels required from 1 to 1022. These password levels are assigned to normal project users.
- **Administrator User Level (1023)** = password level for Administrator or System users. System operations can also be executed With this password level such as closing the project, etc.
- **Developer User Level (1024)** = password level for Developer users. The project can also be developed with this password level.



The User Password Levels are hierarchical meaning that each User of a certain Level can also execute operations of lower levels. Level 1 is the lowest User Level, while the Developer Level is the highest.

### User Level

This property can be set in various points and in various components of a Movicon project. For example, in the **"Users General Properties"** this setting permits each user to be associated with a Password Level.

Where the project's controls and resources are concerned the 'User Level' property is used for defining which User Password Level is necessary for executing the command list associated to that control. For further information please refer to the **'User Levels'** property of the component or resource of interest:

Object "User Level" property (paragraph: "Access Levels Properties common to Drawings and Controls")

Menu "User Level" property (paragraph: "Item Menu General Properties")

Shortcut "User Level" property (paragraph: "Shortcut Command General Properties")

## Access Levels

---

When log on has been executed correctly by verifying the Users authentication it is possible to set a limit to User intervention on various project controls or components by using the **'Access Levels'**.

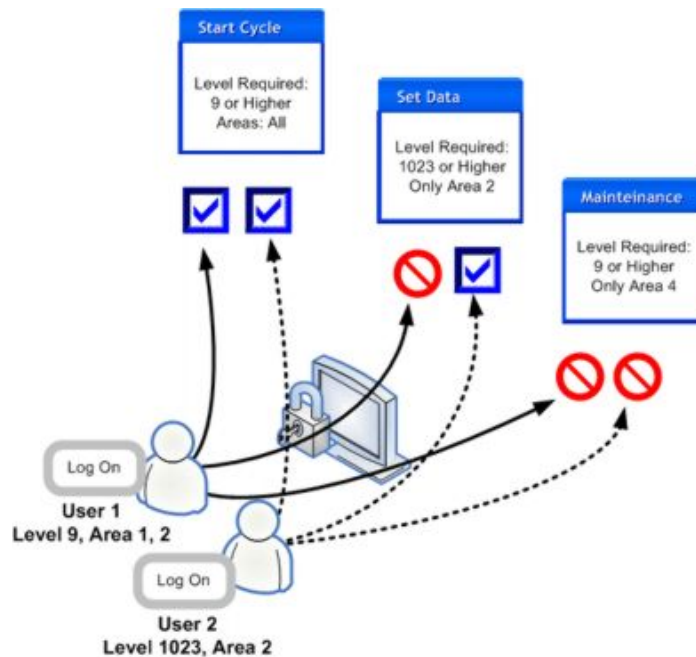
There are 16 distinct levels, from 'Access Level 1' to 'Access Level 16'. In this case each Access Level, which can be activated by using a Check-Box, is independent of the others.

For example, by enabling 'Access Level 5' on a Button control means that only the Users that have 'Access Level 5' checked on their 'Access Level' Masks can access this Button. The Access Mask setup for various Users must therefore have a correspondence with the Access Mask setup for the project's controls or components to be able to grant access, in read/write, to them.



The Access Levels are not managed hierarchically, therefore the high levels (i.e. Level 16) do not acquire the access rights of lower Levels (i.e. Level 1).

You must keep in mind that the **User Password Levels** have priority over the 'Access Levels', which means that the users must first logon and have their Password Level verified for authentication before the Access Levels assigned to the user are verified.



The figure illustrates an example of user access control at Log On. Based on the access level and area required to execute the command, the users 1 & 2 are permitted or denied not access based on their privileges.

#### Write Access Levels

This property, which is found in various controls and components of Movicon projects, defines which is the control's Write Access Level. If the User has been authenticated but their Access Level does not correspond with the control's, the user will not be able to write in that control.

You must also consider that a project's control's or component's write access assumes different meanings according to the object itself. For further information please refer to the **'Write Access Level'** property of the component of interest:

Variable "Write Access Level" property (paragraph: "Variable Access Level Properties")

Alarm Threshold "Write Access Level" property (paragraph: "Alarm Threshold General Properties")

Object "Write Access Level" property (paragraph: "Access Levels Properties common to Drawings and Controls")

Scheduler "Write Access Level" property (paragraph: "Scheduler Access Level Properties")

#### Read Access Levels

This property, found in various Movicon project controls and components, defines which is the control's Read Access Level. If the User has been authenticated but their Access Level does not correspond with the control's, the user will not be able to read that control.

You must also consider that a project's control's or component's write access assumes different meanings according to the object itself. For further information please refer to the **'Read Access Level'** property of the component of interest:

Variable "Read Access Level" property (paragraph: "Variable Access Level Properties")

Alarm Threshold "Read Access Level" property (paragraph: "Alarm Threshold General Properties")

Object "Read Access Level" property (paragraph: "Access Levels Properties common to Drawings and Controls")

Scheduler "Read Access Level" property (paragraph: "Scheduler Access Level Properties")

### Always Visible

This property is only available for the "Scheduler" object and Real Time DB variable resources. This property allows you to keep these resources always available in the project's "Scheduler Window" selection list. By enabling this property the scheduler or variable can be made available and selected in the "Scheduler Window" independently from the Access Level of the user logged in. For further information please refer to the desired component's **"Read Access Level"** property:

Variable "Always Visible" property (paragraph: "Variable Access Level Properties")  
Scheduler "Always Visible" property (paragraph: "Scheduler Access Level Properties")

## Custom Password Dialog Windows

---

Movicon consents custom Password Dialog Windows which open when user authentication is requested or when needing to re-enter an expired user password. Custom panels can actually be created by using the Movicon screens and symbols. The custom panel can be created with a screen within which a "Editable Display" can be positioned with a series of "Buttons". The screen, in order to be managed by Movicon instead of the one for default, must be inserted in the project's "Get Password Screen" or "Expired Password Screen" general properties. This will ensure that the customized screen will open instead of that for default when user authentication is required. The screen will open in modal mode. In addition to the two screens, 'Get Password Screen' and 'Expired Password Screen', manage some local system variables which are needed so that user authentication works correctly. These variables are:

For user authentication, "Password Screen", the local screen variables used are:

**user\_** = variable in which the name of the user must be written and used for requesting authentication. The user\_ variable must be declared string type  
**password\_** = variable in which the password must be written for user authentication. The password\_ variable must be declared string type  
**OK\_** = this variable is needed in the screen's closing phase. When its value is different from zero, the screen will execute the authentication of the user entered when closing. When its value is set at zero, the user will not get logged in when the screen closes. The OK\_ variable must be declared Bit type  
**title\_** = when Movicon opens the Screen the local title\_ variable is set with the requested user level value. the title\_ variable must be declared string type

For entering expired passwords, "Expired Password Screen", the local screen variables used are:

**nMinimunPasswordLength\_** = when Movicon opens the Screen, the nMinimunPasswordLength\_ is set with the password's minimum length value. This values is the same set in the users "Min. Password Length" property. The nMinimunPasswordLength\_ variable must be declared integer type  
**password\_** = variable in which the new password, to be associated to the user, is written. The password\_ variable must be declared string type  
**OK\_** = this variable is needed in the screen's closure phase. When its value is different from zero, the new password will get associated to the user when the screen closes. However, when the value is set at zero, the new password will not get associated to the user when the screen closes. The OK\_ variable must be declared Bit type  
**title\_** = when Movicon opens the screen the local title\_ variable is set with the name of the user changing their password. The title\_ variable must be declared string type



The above describe local variables must be created by the programmer when needed respecting the exact syntaxes. When using Templates from the Movicon Symbol Library, the local variables will be created automatically when inserted on screen.

# General Properties common to Users and User Groups

---

Some properties are available for both User Groups and single User levels. This allows the same properties to be easily propagated to all the users of one group which can then be eventually diversified to differentiate these users. The management of these properties, presented in more than one hierarchy level, provides that the priority starts from the lowest. For example if one of these properties has been defined with single user level settings, these settings will be counted as valid. If, however, nothing has been set for the single user, this property will be inherited by the Group to which that user belongs.

To change the User Group General properties, select the Group from the Project Explorer window with the mouse and use the Movicon **'Properties Window'**.

To check the General properties specifications for each single category please refer to the following paragraphs: "Users and User Groups General Properties", "User Group General Properties", "Users General Properties".

## Style Source Container

You can select the screen to be used as the style source container through this property for managing styles. For further information on the functionality please refer to "Style Sources in Symbols".

## Enable Auto Logoff

A password request will involve the activation of a user in Runtime mode. The activated user will remain active until deactivated in one of the following ways:

1. Deactivate User command (LogOff)
2. Timed Deactivation (Auto LogOff)
3. Activation of a different user

The LogOff commands can be executed by commands associated to the objects or from the project logic.

When this 'Enable Auto Logoff' is activated, the automatic logoff will be executed by Movicon after the time set in the "Auto Logoff Timeout (sec)" property. Countdown will start the moment in which the user discontinues performing operations on the keyboard or with the mouse.



When the 'Enable Auto Logoff' property is disabled, the user who logged on will remain active until the deactivation command is executed (established by the project programmer) or the substituted by a different user.



Movicon allows user activation and deactivation commands to be setup in the project by means of using the command which can be associated to objects or to the functions available in the project's Basic Scripts Logic.

The "Users and User Groups" resource has a tree structure which can be composed of three levels:

**Users and User Groups**  
**User Groups**  
**Users**

The 'Auto Logoff' property can be enabled for each one of these groups. In this case the property will be propagated from the high hierarchical level to those below. For example, if the "Auto Logoff" property is enabled at "Users and User Groups" level this will also be propagated to all the users of all the groups, independently of its group or user level setting. If, however, the "Auto Logoff" is disabled at "Users and User Groups" level, each user will acquire the settings of its own group, and even when it is disabled in its group each single user setting will then be considered.



**Auto Logoff Timeout (sec)**

This property is used for setting the time after which Movicon will execute an Automatic Logoff of the active user. This setting is only significant when the "Enable Auto Logoff" has been activated. The countdown will start the moment in which the user no longer performs operations on the keyboard or with the mouse.

The "Users and User Groups" resource has a tree structure which can be composed of three levels:

**Users and User Groups**  
**User Groups**  
**Users**

A value can be assigned to the "Auto Logoff Time (sec)" of each one of these levels. In this case the value associated to the lowest hierarchical level to which the "Auto Logoff" property has been enabled will be considered. For example, if the "Auto Logoff" property has been enabled on all three levels, the Logoff time for each user will be the one specified in the property of each single user.

**Logon Script**

This selection allows you to choose a Basic Script routine to be executed during the User Logon stage. When this setting is executed in the User's properties, the Basic Script routine will only be executed when the specified user logs on. When, however, the setting is executed in the property of a User group, the Basic Script routine will be executed when all the users belonging to that group log on.

**Logoff Script**

This selection allows you to choose a Basic Script routine to be executed during the User Logoff stage. When this setting is executed in the User's properties, the Basic Script routine will only be executed when the specified user logs off. When, however, the setting is executed in the property of a User group, the Basic Script routine will be executed when all the users belonging to that group log off.

**Logon Commands**

This selection allows you set a "Commands List" to be executed during the User Logon phase. When this setting is executed in the User property, the "Commands List" will be executed only when the specified User Logs on. When the setting is executed in the Users Group property, the "Commands List" will be executed when all the Users belonging to the Group log on.

**Logoff Commands**

This selection allows you to choose a "Commands List" to be executed during the User Logoff phase. When this setting is executed in the User property, the "Commands List" will be executed only when the specified User Logs off. When the setting is executed in the Users Group property, the "Commands List" will be executed when all the Users belonging to the Group log off.

**Expiring Password (Days)**

This property is used for setting the number of days after which the User's Password will expire and will no longer be usable. Once expired a new pass will be requested for the next user authentication. When this value is left at zero on the User's property then will be considered the Group's property value. When this value is left at zero in both the two properties then the password will not expire.

**Language**

This property allows you set a default language for the User. When this setting is executed in the User properties, this default "language will only be valid for that User. However, if set in the User Group properties, the default "Language" will be valid for all those users belonging to that User group and which don't have the same property set.

When inserting a language in this property, it will automatically activate when the User logs on. This consents to you to have a multilanguage system where each user can be associated a different language for default as desired.

# Users and User Groups Properties

---

You can setup Users and User Groups in projects, which are needed for either sending messages on event or alarm, or for managing access security to project functionalities. Each command which can be setup by Movicon in the project can be associated with a password request, in relation to user level or a specific user. In addition to this the Movicon system commands require (if the password management is active) an access level equal to or higher than the "Administrator" level 1023.

By using the **"Users & User Groups"** resource you can set the project's security properties, such as activating the password management in the project, installing users so they can access the operation and system performances in function with the password level they have been associated with, etc.



The password will be requested only during the project Runtime when the "Enable Password Manager" property is enabled before its startup. The password with Developer level will be also requested when opening the project in programming mode where the "Password Protected Project" property has been enabled. Therefore it is strongly advised to safeguard the "Developer" level password and not forget it.

The project Users can also be inserted (or edited) during the project runtime, by using the purpose built RunTime Users editing functions, if the Project's "Enable Runtime users" property has been enabled.



The Users which can be inserted during the RunTime cannot receive in associated the "Developer" level (level 1024), reserved for the programmer.

## Users and User Groups General Properties

---

By using the Users and User Groups General properties you can define how to manage project protection. For instance, you can enable the protection on project editing, active the password management in Runtime, etc.

To modify the Users and User Groups General properties, select "Users & User Groups" from the Project Explorer window with the mouse and use the Movicon **"Properties Window"**.

To verify the General properties common to Users and User Groups please refer to the "Users and User Groups General Properties" paragraph.

### Enable Runtime users

This selection allows you to activate the Runtime Users management so that new users can be added, changed, deleted during project runtime. In order to execute these functions the appropriate commands in the Commands list of the Movicon controls or the dedicated Basic Script function must be used.



**This property is only managed when the "Enable Password Manager" property has been activated. Otherwise the Runtime users will not be managed by Movicon.**

### User Level Editable

This property allows you to define the User Level where the ones above will not be able to carry out any editing in Runtime. For example, if "User Level 5" is set, this means only users up to the fifth level can carry out any editing during Runtime.

### Password Protected Project

This selection permits you to activate project protection by using passwords. When the protection is enabled only users with Developer rights (level 1024) can open the project in development mode.



Keep in mind that a project with active protection will not request any passwords at the system startup in Runtime, but the Developer password will be required (Level 1024) for entering in programming mode or for opening the project in programming mode.

Before activating the password protection, the system will ask that there be at least one user with Developer password (level 1024) present.

**WARNING! If the User Name of Developer Password is forgotten, it will not be possible to open the project in programming mode.**

#### Enable Password Manager

This selection permits you to activate the project's password manager during Runtime phase. By doing this you can associate different password levels to the project commands according to their requirements.



When the project's password management is enabled the System's operations (or Administrator) such as closing the project, can only be done by Users with **Administrator** rights.

#### Enable CFR21- Part 11 Settings

This selection permits you to set User's Description like Electronic Signature in conformity with the **"FDA21 CFR Part 11"** normative.

#### Set CFR21-Part 11 Settings

This selection permits activation of User settings editing in conformity with the **"FDA21 CFR Part 11"** normative.

#### Min. User Name length

This property is used for setting the minimum number of characters which must be used for composing the Name of Users setup in the project.



Four characters is the minimum length set by Movicon for default. We suggest you don't use less than this for normative and security reasons.

#### Min. Password Length

This property is used for setting the minimum number of characters which must be used for composing the Password of users setup in the project.



Six characters is the minimum length set by Movicon for default. We suggest you don't use less than this for normative and security reasons.

## Users and User Groups Windows User Properties

---

By using the Windows User properties you can enable the option to use Operating System users inside your Movicon project. This will also allow you to associate user levels to the O.S. user groups.

To change the Windows Users properties, select the "Users And User Groups" group from the Project Explorer window with the mouse and use the Movicon **"Properties Window"**.

#### Enable Windows Users

Movicon provides you with the possibility to share users from the O.S. domain or from a Windows Server in the applied project.

In this way, when the password management is activated, the project will acknowledge and grant access to users inserted and activated from the O.S. domain installed or a Windows server.

Movicon accepts mixed applications, such as users inserted in the project list and users deriving from the domain.



The users from the project list can receive in association a customized user level. The users deriving from a domain of a Windows station can receive a customized user level only if they have been inserted in the users list, otherwise they will be associated the same password level specified for the group they belong to (Administrators, Users, Guests).

Different password levels can be assigned to domain users. This function is available when adding users who have the same userID configured in the primary domain controller, to the Movicon users list. The authentication of a user who has been configured in this way, is carried out by the primary domain controller which controls the validity of the password.

For instance, if a user exists with UserID = "guest" inside the primary domain controller, the user should then be configured in the Movicon project's user list with the same UserID, 'guest' with the password left empty so that the user can be assigned with the level desired. In runtime the user name and password inserted in the Movicon Authentication window are validated by the primary domain controller. This also permits the password expiration to be used with Windows users.

This mechanism is also valid for users configured directly in runtime with the Movicon users editor window.

#### **Windows Admins Level**

This property is used for selecting the Password level to assign to the Operating System Users belonging to the Administrators group.

#### **Windows Users Level**

This property is used for selecting the Password level to assign to the Operating System Users belonging to the Users group.

#### **Windows Guests Level**

This property is used for selecting the Password level to assign to the Operating System Users belonging to the Guests group.

#### **Windows Admins Access Level**

This property is used for selecting the Access level to assign to the Operating System Users belonging to the Administrators group.

#### **Windows Users Access Level**

This property is used for selecting the Access level to assign to the Operating System Users belonging to the Users group.

#### **Windows Guests Access Level**

This property is used for selecting the access level to assign to the Operating System Users belonging to the Guests group.

## **Users and User Groups Child Project Option Properties**

---

By using the Child Project Option properties you can enable the possibility to use eventual declared Child Projects internal the Movicon project.

To edit the Child Project Option properties, select the "Users and User Groups" group from the Project Explorer window with the mouse and use the Movicon **"Properties Window"**.

#### **Include Child Project Users**

When this property is enabled users of eventual Child Projects setup in the Project will be inherited. This makes it possible to enable and use child project users in the parent project according to their password and access levels.



**This property will only be managed when the "Enable Password Manager" property has been activated. Otherwise child project users WILL NOT be inherited from the parent project.**

#### **Child Project User Max Level**

This property permits you to set the maximum password level which can be inherited by a child project. The users in the child project who have a higher level to that set in this property will not be inherited and therefore cannot be used by the parent project.

# User Group Properties

---

The Movicon User management provides the possibility to put project users into groups.

Not only does the User Group setup clearly organize the plant's users, but also gives you the great advantage of being able to send SMS, Fax or voice messages automatically to preset user groups on the list. When creating a new project with the "Create Default User Groups" option enabled, Movicon will insert four default groups, "**Developers**", "**Administrators**", "**Users**" and "**Guests**". Apart from the "Developers" group, all the other three reflect the classic O.S. groupings. You can then create customized groups where the project Users can be inserted and then edited. Each User Group can be associated with properties, such as password levels, which can be propagated to all the users belonging to the same group. You can also customize further each single user of the Group. To modify the User Group Properties by selecting the Group in question from the Project's Explorer Window with the mouse and using the Movicon "**Properties Window**".

## User Group General Properties

---

The User Group General Properties are used for associating each group with the main characteristics concerning the password and access levels for the group's users.

To modify the User Group General Properties, select the Group from the Project Explorer window with the mouse and use the Movicon "**Properties Window**".

To check the General properties common to Users and User Groups please refer to the paragraph on "General Properties common to Users and User Groups".

### Name

The name of the group you wish to create is entered in this edit box.



When creating a new project Movicon will insert four default groups: "**Developers**", "**Administrators**", "**Users**" and "**Guests**" if the "Create Default User Groups" option has been enabled. Apart from the "Developers" Group, the other three reflect the classic O.S. groupings.

### Description

The Group's description is entered in this edit box. The description is only used as a reminder for the programmer and appears in the Group's "**Properties Window**" only.

### Default Level

The Password Level to be associated to the Group is set in this property. This property will also be propagated automatically to all the Users created within this group, until any later modifications are made through these Users' properties. The password levels associated to groups created for default by Movicon are:

- **Developers:** Developer Level (Level 1024)
- **Administrator:** Administrator Level (Level 1023)
- **Users:** User Level 5 (Level 5)
- **Guests:** User Level 0 (Level 0)

### Default Access Level

The Access Level mask to be associated to the Group is set in this property. This property will also be programmed automatically to the Users created within this group until any later modifications are made through these Users' properties. The Access Levels associated to the groups created for default by Movicon are:

- **Developers:** FFFF (Access Levels 1-16)
- **Administrators:** FFFF (Access Levels 1-16)
- **Users:** 0010 (Access Levels 5)
- **Guests:** 0000 (Access Levels 0)

# Users Properties

---

The Movicon Users management, in addition to running projects with security, gives you the great advantage of sending SMS, Fax or voice messages automatically to one or more preset users in the list. This is due to possibility of associating each client with a customizable telephone, e-mail and FAX number etc. When a new project is being created Movicon will insert the users belonging to the Operating System's domain if the "Create Users from Windows Users -> Server Name" is enabled. These Users will automatically be inserted into the relative "**Administrators**", "**Users**" and "**Guests**" groups if the "Create Default User Groups" has also been enabled or otherwise inserted in one single group for default.

To edit the Users General properties, select the User in question from the Project Explorer window with the mouse and use the Movicon "**Properties Window**".

## Users General Properties

---

The Users General Properties are used for associating each user with the main characteristics concerning the password and access levels for those users.

To edit the Users General properties, select the User from the Project Explorer window with the mouse and use the Movicon "**Properties Window**".

To check the General properties common to Users and User Groups please refer the paragraph on "General Properties common to Users and User Groups".

### Name

The user's name (User ID) is assigned in this edit box. The name will be the first information requested as soon as the command dealing with the password is executed in Runtime. Alphanumeric and case sensitive characters, discrimination between upper/lowercase keys, can be inserted in this field.

The user name must be unique and obligatory.



Four characters is the minimum length, imposed by Movicon for default, for the User's Name. We suggest that you do not use less characters for normatives and security reasons.



When creating a new project with the "Create Users from Windows Users -> Server Name", option enabled, Movicon will insert the users belonging to the Operating System's domain. These Users will automatically be inserted in the relative "**Administrators**", "**Users**" and "**Guests**" groups if the "Create Default User Groups" option has also been enabled, otherwise they will all be inserted in one group only for default.

### Description

A descriptive text of the user's profile can be associated to the user's name in this edit box.

The user's description will be used by the system to identify the active User, and then registered where needed when the electronic signature is required.

The description is unique and obligatory.

### Password

This property is used for entering the Password relating to the User. Alphanumeric and case sensitive characters, discrimination between upper/lowercase keys, can be inserted in this field.

Password declaration is obligatory.



Six characters is the minimum length, imposed by Movicon for default, for the User's Name. We suggest that you do not use less characters for normatives and security reasons.

### **Numeric Password**

This property allows you to insert a numeric Password which can be used as an alternative to the alphanumeric one.

### **User Level**

This property is used for setting a Password level to be associated to the User. The field is set at **"Get Group Level (-1)"** for default, which means that the User automatically acquires the Level from the Group it belongs to. However a customized level can be selected for the User, independently of the Group it belongs to.

Movicon provides up to 1024 password levels, where the first 1022 levels are inferior type (for users), level 1023 or "Administrators" also authorize commands in the operating system, level 1024 or "Developer" is the highest and authorizes access to the project being programmed.

Level declaration is obligatory.

### **User Access**

This property is used for setting the Access Level Mask to be associated to the User. When creating the User this property is set automatically to the same value of the **"Default Access Level"** of the owner group.

For further information on "Access Levels" see paragraph **"User Levels and Access Levels"**.

### **Disabled**

This property is used for disabling the User. This means the user will not be managed in the runtime phase.

### **Locked**

This property locks the User out. the 'Locked' user is one who cannot be substituted with a runtime user who has the same name. This means that any runtime user with the same name will not be managed. For further information please refer to the section on **"Runtime Users"**.

### **Must Change Password**

When this property is enabled the new created user will be asked to change their password when logging on for the first time. This will allow the user to change the password, inserted by the programmer, with their own.

## **User Data Properties**

---

The User Data properties are used for associating each user with a different telephone number or e-mail address to which SMS, FAX, e-mail, etc., can be sent.

To change the User Data properties, select the User from the Project Explorer window with the mouse and use the Movicon **"Properties Window"**.

**Note: This is an optional feature: check your dongle options.**

### **Mobile Country Code**

The country code relating to the User's mobile number is entered in this box.

### **Mobile Area Code**

The area code of the User's mobile number is entered in this box.

### **Mobile Phone Number**

The User's mobile phone number is entered in this box.

### **Voice County Code**

The country code relating to the User's voice message number is entered in this box.

### **Voice Area Code**

The area code relating to the User's voice message number is entered in this box.

### **Voice Phone Number**

The User's voice message number is entered in this box.

### **Fax Country code**

The country code relating to the User's Fax number is entered in this box.

**Fax Area Code**

The area code relating to the User's Fax number is entered in this box.

**Fax phone Number**

The User's Fax number is entered in this box.

**Messenger Recipient**

The name or address of the User's Messenger Recipient for sending messages exploiting the MSN Microsoft Messenger program is entered in this box.

**E-mail**

The User's E-mail address for sending e-mails is entered in this box.

## RunTime Users Manager

---

The Movicon Users management also provides the possibility to edit, change or add users during the project Runtime mode. The "EditUsr.exe" tool is used for this purpose which is installed with Movicon and resides in the Movicon installation folder. This tool can be executed by using the **"Edit User List"** command found in the Movicon **"Command List"**.

The Users created in Runtime are saved in the "ProjectName.rtusers" file which is saved in the project folder in un formato criptato.



The runtime users are saved in crypt mode starting from the "EditUsr.exe" tool's 1.0.0.8 version. In this way their information can not be read by external programs, due to the fact that the application crypt key is used. This new version of the runtime users editor can be used with any Movicon version.

If this file is present, Movicon will also load the Runtime users in this file at the project startup. In addition to this, this file will be reloaded during Runtime each time it is changed with the "EditUsr.exe" tool.

The Runtime Users are therefore Users who can be used within the project like the user created in the Development mode are. However you need to take into account the following rules when managing runtime users:

1. The loading Runtime users will be only be executed when the **"Enable Runtime User"** property from the **"Users and User Groups"** resource has been selected.
2. Movicon, when loading the file containing the runtime users, will only activate those users with a level not higher than the one set in the programming mode in the **"User Level Editable"** property from the **"Users and User Groups"** resource. A warning message will be generated in the Output and Historical Log Window for those users with a higher level who will not be activated
3. Movicon, when loading the file containing the runtime users, will not activate those users who have the same **"Description or Digital Signature"** of a already existing user. In this case a warning message will be generated in the Output and Historical Log window
4. When a runtime user is created with the same name of user declared in the programming phase, they will be replaced by the runtime user. In this way the properties of the user created in the programming mode can be changed during runtime. The settings of the user created in the programming mode will not be actually changed just simply those settings of the runtime user will be loaded. This mechanism can be executed only when the **"Locked"** property of the user, created in the programming mode, has not been enabled. Otherwise the runtime user will be refused generating a warning message in the Output and Historical Log window
5. When a runtime Group is created with the same name of a Group declared in the programming mode, this group will be replaced with the runtime Group

The "EditUsr.exe" tool can also be executed independently of Movicon. For further information on how the "EditUsr.exe" tool works, please refer to the application's help.



## Virtual Keyboard Use

---

The Runtime Users Editor provides the option to use an alphanumeric pad with the same look as the standard one used in the Movicon application. A button has been purposely placed at the side of each edit box so that it can be called whenever needing to edit values for systems with touch screens.

This function has been inserted both in the desktop version and that for Windows CE. The difference between the two is that the Windows CE version still has the option for you to use the operating system virtual keyboard (SIP) by simply clicking on the editbox. In addition to this some of the "General" key values are managed in the Windows CE version as well:

- "AlphaNumericPadFont" and "AlphaNumericPadFontSize" to choose the font type and font text size for the alphanumeric pad. This will change the size of the pad window as well according to the options chosen.
- "ShowSIP" to enable or disable the appearance of the Windows CE virtual keyboard with a click on the runtime user editor' editable controls.



Attention: the runtime user editor is no longer supported and installed for PocketPC platforms.  
It is only available for Windows XP/VISTA and Windows CE in SSDK devices.

# CFR21 Part 11 General Concepts

---

The aim of the CFR21 Part 11 regulations, written up by the FDA (Food & Drug Administration), is to obtain a legal equivalence between electronic documents (digital records and electronic signatures) and traditional paper documents. This has evolved due to the increasingly frequent use of automatic systems in managing production processes in systems that operate under FDA approval. In order that automation and control systems are realised in conformance with the CFR21 Part 11 regulations it is necessary that all recorded data is made referable to the operator in charge (Electronic signature), furthermore certain regulations regarding any precautions must be adapted to safeguard against forgery and mishandling of electronically recorded data, or to allow easy identification of any misuses, whether intentionally or unintentionally, of electronic devices which generate electronic records. Many pharmaceutical industries have especially benefited from using electronic records where untold amounts of paper documentation, archived over many years of research, has been transferred into electronic records which not only has reduced space but also precious time in acquiring and reviewing important information before releasing medicine on the market for sale.

It is absolutely crucial that these types of industries have the devices with the right protection mechanisms to safeguard against any intentional or unintentional data errors in electronic format.



## General Concepts for supporting these regulations

---

The concepts described below define how to use Movicon to develop applied projects with compatibility with the act and its regulations discussed in this document.

A list of the main concepts has been put together by Progea to give a clearer picture on the indications explained henceforth and which are based on the understanding that it remains the user's responsibility to ensure that the application, developed with Movicon, is compliant with these requirements.

## Security

---

- The Movicon project must be encrypted (Movicon uses a 128 bit encoding) so that all the configurations and passwords used in the project are accessible from the outside.
- Movicon guarantees unique user password entries in the project. Each user is identified in the project with a UserID, Password, printable Description or Name (Electronic Signature). Movicon does not accept Users with the same electronic signature name (unique identity control) of another individual. The names must be made up with not less than 4 characters and not more than 64 characters.
- To guarantee data integrity and safeguard against any tampering of data, the Movicon application should be run as Service of the Windows operating systems. This will require identification of users registered in the system's domain according to the security requirements stipulated in order to access the operating system and its files.
- Movicon supports Windows OS domain sharing so that the user passwords, set up by the system administrator, can be used.
- Users who manage the recording of data by using the Data Loggers must take the right measures to prevent any unauthorized access, undesired modifications and tampering to database records. The IMDB archives (InMemory DB) allow users to manage encrypted historical log files or secure databases can be used, such as Microsoft SQL Server or Oracle with the appropriate administering of the Win2000/XP operating system, which only permit the system administrator or developer access to records.

- To put an access limit on the developed application's functions and controls, the Movicon project must use the User Password Profile management correctly, which involves the entering of a Password, UserID, User Name and Access Level. Movicon provides 1024 access levels and 16 areas.
- Users must manage their passwords with great care and integrity. New users, inserted by the administrator, can replace their password with a more personal one on their next Log On.
- All passwords can be set with an expiry time to make the user to issue a new password periodically, which will contribute to increasing system security.
- To fully comply with the regulations, the Auto LogOff (timeout of enabled access) must be appropriately used in the Movicon password management in order to prevent unauthorized access to the system after a certain period of user inactivity.
- To ensure validity and the correct entering of data, users must make sure that the Movicon operating stations are allocated in safe places and that they are accessible to authorized personnel only.
- The Movicon AutoLogoff function must be used in systems in continuous use.
- Movicon has tools and procedures that can be used for discouraging any unauthorized access attempts and are the same as those used in the Win2000/XP operating system as required by the regulations. After the third failed attempt to access, Movicon will purposely take longer to respond to the re-entry of the password to discourage the intruder.
- Any further attempts to violate the system (Upon the fifth unauthorized Log On attempt) Movicon will display and record the event in the Historical Log in order to safeguard against and control any further system violations.

## **Digital Recording/Electronic Signatures**

---

- Movicon returns the descriptive name of the registered user to identify and individualize the active operator.
- The applied program must be configured to record electronic signatures each time a digital recording is carried out (creating a record in the database) as required by the regulations. The user must execute LogOn in the project by linking two combined data (UserID and Password), and the electronic signature must be the genuine name of the user, the date, time and reason for the recording. The Movicon Data Logger allows the recording of all necessary data on the Database.
- For reasons of legal responsibility relating to the Electronic Recording, the operator must always be acknowledged every time data is recorded or when accessing the system. The User's ID is unique and belongs to that user only in Movicon and no other individuals are allowed the same ID.
- To satisfy the Electronic Recording requirements, the recording of events must be configured appropriately by using the IMDB archives (InMemory DB) where crypted historical log files can be managed or if ODBC archives, such as the Microsoft SQL Server or Oracle, secure databases must be used with the correct security management administered. Furthermore the user must configure applications to acquire and record electronic signatures on record of any operator undertaking actions. The user must also prevent any data from being lost by configuring the application to execute backups of all data recorded, or by using the Movicon redundancy functions. The user can also eventually configure the system so that it uses the Data Logger resource to record crypted data on IMDB or on relational ODBC database files. If needed, new data files can be created with prefixed timeframes (eg. Every 4, 8 or 24 hours) by using the Basic Script functions.
- The user can configure the system to copy recorded data in a safe and secure location by using procedures appropriately written with Basic Script codes. The Windows OS security functions protect files and their data from any unauthorized access. When multiple files are created the user must control whether the right code is entered to prevent saturating free space on the hard disk where the oldest files may need to be deleted.
- The user may have to synchronize the system's time in real time or to that of another system's (Microsoft or third parties) so that recorded data relate to the true date and time, or they may have to manage data synchronization between Client and Server so recording becomes coherent. Synchronization of this type

can be managed directly with the Windows OS functions or with the Basic Script codes for third party products.

## **Validation and Documentation**

---

- Some of the requirements stipulated in the regulations are not altogether implemented in software applications. These Part 11 requirements can be satisfied if the client validates their application to guarantee accuracy, reliability and security when recording data, as well as the capacity to prevent unauthorized editing, errors and data deletions. The Movicon users must validate their application in order to comply with the FDA act. The users can develop and/or execute the validation of programs and protocol themselves or delegate this task to others. The validation must follow a methodology established from system's life cycle (SLC).
- In order to meet the controls requested by the regulations in this act, the client must adopt adequate procedures to verify the identity of the individuals who have been assigned an electronic signature.
- The client must enter and set up the operator and their operating responsibilities executed under their electronic signature, to impede any forgeries or tampering of signatures or recordings, in compliance to the regulations of this act.
- The client must always be certain on the identity of the individual assigned an electronic signature. Further more the client is held responsible that the enrolled operator is fully aware of the regulations stipulated by the FDA agency and that they intend to use their electronic signature as a substitution and an equivalent of their own handwritten signature used on traditional paper and, when necessary, produce certification of their true identity, being legally binding to their handwritten signature, when under FDA inspection.
- The client is responsible for producing documentation on system use or on the application realized, on its distribution and updates, and also the details on personnel training. However, the client is not responsible for documentation on the platforms being used (Movicon, Windows).

## **Other**

---

- All the data must be stored in a relational database, which fully meets the necessary security requirements (ie. IMDB crypted data, SQL Server or Oracle with the relevant protection) and protected from any violation to or tampering of the security functions belonging to the Win2000/XP OS. Data must be filed and kept available for an adequate period of time according to the operating requirements.
- To further enforce the safeguard of data, project, images and recipes the user should use a third party software type, which can guarantee version maintenance and management (eg. Microsoft Source Safe can be used for controlling the versions).

# **Validation Configuring Techniques**

---

To get a Movicon project 21CFR Part 11 ready, you need to configure it appropriately so that it is compliant with the FDA validation criteria. The necessary measures to take in doing this are indicated below:

## **Security**

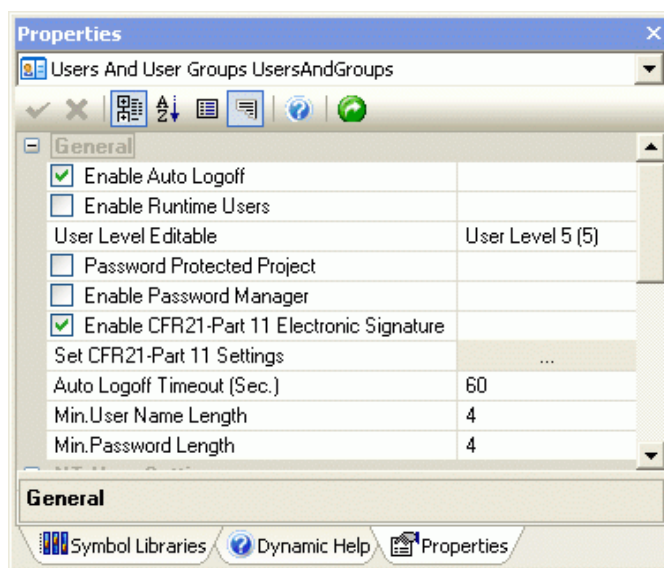
---

- The project must be configured in its General Properties by selecting "Crypted Project" and "Crypted Project Resources". In this case all the project's XML information can be accessed by using a 128 bit encoding.
- To prevent unauthorized system access, select all the project's Execution Properties which deny Operating System and Desktop access. The following can be denied:

1. Windows Desktop
  2. The Start button form the Windows' Task bar
  3. Windows Task Bar
  4. Windows task Options
  5. Windows Task Manager
  6. Windows CTRL+ALT+DEL
- As an alternative, Movicon can be run as "Service" of the operating system, as described in more detail ahead. When Movicon is run as service, it does not manage the project's properties that deny operating system access and which are listed above.

## Passwords

- All the application commands that can be executed by operators to interact on the process must be protected by passwords.
- The password management must be enabled in the project's User Passwords resource Properties:
  1. Project Protected with Password: the password will be requested only for entering in "Development" mode
  2. Enable Password Management: the passwords will be activated according to the levels and access modalities to the preset commands
  3. Enable Electronic Signature: the unique user Description of the user whose name is to be used as an Electronic Signature will be managed
  4. Auto Log Off: determines the time (sec.) for automatically deactivating the active user after a period of inactivity
  5. Minimum Length (user name and password): set for default at 4 and 6 characters respectively, as suggested by the regulations
- Secondary parameters relating to the password management need to be set according to the general properties illustrated below



## Windows Security Synchronization

Movicon provides the possibility to share, in applied applications, users from the operating system domain or from a Windows server.

Therefore, when activating the password management, the project will acknowledge and accept users inserted and activated from the domain of the installed operating or from a server station.

Movicon accepts mixed configurations, whether being users inserted on the project list, or users deriving from the XP domain.

Users on the project list can be associated with a customized user level. Users deriving from a Windows station domain can receive a customized user level when inserted on the user list, otherwise they will be associated the same password level specified for the group they belong to (Administrators, Users, Guests).

Different password levels can be assigned to domain users. This function is made available by adding users to the Movicon user list who have the same UserID configured in the primary domain controller. Password authentication and validity are carried out by the primary domain controller for those users configured in this way.

For instance if a user with UserID = "guest" exists in the primary domain controller, a user can be configured with the same UserID, in this case "guest", and the password can be left blank in the Movicon project's user list. By doing this the user can be assigned the appropriated level desired. In runtime, the user's name and password, which are inserted in the Movicon user authentication window for logging on, are validated by the primary domain controller. This permits expiry passwords to also be used for Windows users.

This feature is also valid for users configured directly in runtime with the Movicon edit users window.

## **User Passwords**

---

- Each user or user group who has access to commands or process interaction, must be inserted and configured in the project appropriately.
- Users are inserted in the project's User Password Resource where they can be configured in their properties. These properties include those which involve the requirements stipulated in the FDA act:
  1. Name (ID) and Password: These are assigned to the user and are used for identification by the system
  2. Electronic Signature: This is a unique text which corresponds to the user's electronic signature and is recorded as absolute user identification (the Electronic Signature management must be enabled in the User Password Resource)
  3. Auto Log Off: This can be specified singularly for each individual user
  4. Expiring Password: The act stipulates that the user password expires after a certain preset time so that the user is obliged to change it periodically to increase system security
  5. Must Change Password: For identification certainty this obliges the user to enter their own personal password on the next Log On so it is only known to themselves and no one else including the administrator who logged them on the first time

## **Command Access**

---

- Each command, change or setting influencing the process must be given protected access by requesting user identification.
- The User Level in a hierarchical scale structure must be set in the "Access Level" property of each object. The Levels in Movicon start from 1023 (reserved for the system administrator) to level 1 (the lowest operating level). The 1024 level is reserved for the programmer.
- The command objects can also be provided with a Access Level (Area) in read or write, permitting users to access commands not only on a hierarchy scale but also by area of competence.

## **Operating System Access**

---

Movicon provide the possibility to block and deny operating system access. The follow two modalities can be used:

- Lockout Windows access from Movicon: to prevent unauthorized access in the system you need to select all the project's Execution Properties which deny access to the Operating System and Desktop. When Movicon is started up these will deny access to Windows according to the settings, which have been activated (described above).
- Using Windows services: you can configure the project to be installed as the operating system's Service, therefore it will be started up automatically before

the Windows operating System's Log On procedure. By doing this, as an alternative to the above indications, only the system administrator can access the operating system.

## Biometric Systems

---

Using Biometric Systems is highly recommended in application validity according to the regulations.

In this case you need to choose the right recognition system among those available on the market that can be easily integrated into your application.

The most popular biometrics systems are ultimately those that use digital fingerprints. These systems are simple to use and integrate perfectly with operating systems and software applications.

Examples:

- Progea has run tests on the Toca Fkey product (digital fingerprint scanner). This device can be plugged in to a USB port and has its own user profile management where the Movicon project users can be associated by using the appropriate VBA script module provided by Progea. This biometric system can be completely integrated into the project using the Movicon "User Password –Fingerprint" association.
- Progea has also run tests on the Microsoft Fingerprint product, a simple and reasonably priced device that can be plugged into any USB port with Windows XP. This system runs its own software as service and provides files where users are inserted and recognised by their biometrics every time a password entry request is made. A tool, such as this one, does not require any project modifications or any particular interfacing or configuration. However, authentication of the operating system's users (Windows XP/Vista only) is only allowed when the PC users do not belong to a Domain.

Any type of biometrics recognition system can easily be used if the operating system has been predisposed to support one as described above, otherwise it can be integrated into the Movicon application by using the appropriate Basic Script interface.

## Recording Data (Audit Trail or Tracing)

---

Movicon provides the possibility to trace all the status changes of each variable which has significant relevance to or influence on the process: Usually all the set-point or process command changes need to be traced.

- Note the difference between the Trace and Data Logger files: The Trace records each data value change in the appropriate database along with all the relevant information, while the historical value recordings refer to the historical logging activity executed by the Data Logger resource.

In certain cases, it is sufficient enough to carry out the following procedures to sensitive data:

1. Request user identification before accessing to commando
2. Identify user and validate them (password management)
3. The user carries out the changes. The variable (Tag) is traced
4. The value change is recorded in the appropriate Trace DB, reporting the date, the previous value, the current one and electronic signature

All the historical information inherent to each change that took place in the process can be obtained from the appropriate Trace viewer so it can be easily traced back to what caused it.

The Tracing function is one of the properties belonging to each single Variable (Tag) and must be activated and configured by clicking on the "Trace Options" property in each Tag (variable).

## Audit Trail

---

In many cases, before the user can proceed in making any process variable changes (eg. Set points), confirmation may be requested before the change can be put into action,

together with a comment to explain the reason why this change has been made (text string). In order to enter this comment the "Trace Comment" item needs to be ticked/checked in the Trace Property beforehand.

Movicon will display the window shown above after each manual Tag change occurs and authenticated by the user, indicating the change and requesting the user to state the reason this change was made. The comment inserted by the user is recorded:

- In the 'ActionCol' column of the Tracing DB table referring to variable which was changed
- If the 'Add Msg to SysLog' check box has been checked, the event and the comment are also recorded in the main historical Log DB, in the 'DescCol' of the Historical Log's 'SysMsgs' table
- Note: When the 'Trace Comment' window is open on screen, the variable's value is frozen. Any other process, such as the drivers, the IL logic, basic scripts, cannot change it

## **Audit Trail with Process Manager Validation**

---

There maybe times when the above described operations need not only the operator user's authentication but also validation from the Process Manager before a Tag change can be made effective. However, authentication must only be requested from Process Managers with the same level or higher.

As each process has different needs from the next, Movicon does not manage this function automatically the user must provide a Template being a graphic object that can be called up every time an edit request is made. This object allows access, user identification and data settings (Tag variables), which can be linked to both Tracing function and a Data Logger which have been configured to record the values relating to each status change.

## **Electronic Records**

---

Electronic Records contain all the process information (dates, values, events) recorded electronically on files that must guarantee data integrity and prevent any unlawful handling from unauthorized persons.

All the information recorded on file by Movicon is called "Electronic Records".

In order to get the Movicon Electronic Records standard ready, the following indications and the guidelines contained in this document need to be followed to guarantee security in data integrity and prevention against any unauthorized access and data tampering.

## **Data Security**

---

Guaranteed Electronic Record security is absolutely fundamental in obtaining valid applications. The data recorded by Movicon (Data Loggers, Log, Tracing) are physically built by:

- IMDB: criptable XML text files with an algorithm in 128 bits. To use this format you need to check the "Crypt File" option to guarantee inaccessibility to external manipulation of historically logged data.
- ODBC: Relational Databases by means of the integrated ODBC manager. The data, therefore, physically resides in data files and tables that can be recorded on hard disk locally or on mass files residing physically in diverse servers. Thanks to the use of "safe" relational databases such as SQL Server, Oracle or others, Movicon uses protected accounts for accessing files. It is the user's responsibility to configure the system so that no one can access files, by removing access rights to file both in the database itself and in the operating system folders access rights (Movicon run as service). Data security must be guaranteed by means of using the following procedures:
  1. Always use a data format based on relational databases that provide access protection, such as Microsoft SQL or Oracle
  2. To avoid unauthorized access to files, User Account protection will need to be setup by using the access criteria explicitly for system administrators or program designers only (eg. With the same project protection password). This will impede access to data tables where authorization has not been provided.



3. Use the operating system's access lock (Locked by Movicon) or access rights to operating system by using Movicon as Service. By doing this, file access through the operating system will be physically denied
4. Do not share folders or disks when the station is operating in net, except for system administrator access
5. Remove all rights to modify database records (Updates). Movicon lets new records to be inserted whose data cannot be accessed for altering no matter what the reason is



A special document describing this normative and its regulations in relation to Movicon projects has been published by Progea. Please contact the Progea Support Center if you need this document.

## 22. Data Loggers and Recipes

---

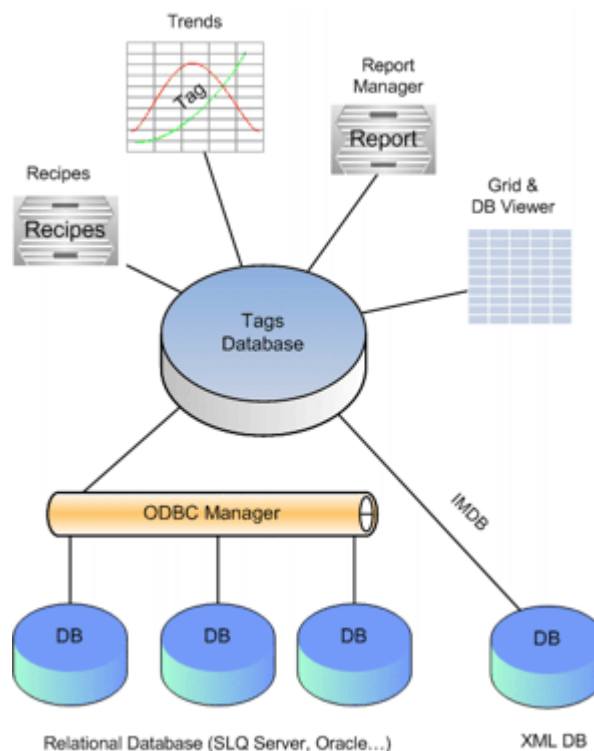
***The Data Logger is a powerful tool for recording and simplifying the managing of data in databases, both ODBC and XML text. The Reports, Trends and Recipes functions can all be linked to the Data Logger.***

This section describes the **Data Loggers & Recipe** resource, which is the main tool for recording process data (Variables or Tags) on archive files, with the aim of creating and managing historicals which can be used for executing analysis in the project.

The project's Data Loggers and Recipe resource can contain an unlimited number of Data Logger and/or Recipe objects, which have the task to sample associated variables (Columns) and record data on file according to the recording modes setup in the project's properties.

The archive management and input/write modalities of the file structures are described in the "Historical Log Management" section where the possibility to use ODBC files in a relational database, and use IMDB files in XML test.

All the data managed by Data Loggers will be available in the project in Reports, Recipes, Charts or in any other display functionality, as well as analysis on stored process data as illustrated below:



Before using a report or data table, you will need to configure the database according to your requirements.

You can access the Movicon database engine settings through the "**Data Loggers & Recipes**" resource which can be selected from the "**Project Explorer**" window.

The Movicon Data Loggers and Recipes can access Database files through **ODBC** or **IMDB** links, where the user can get production data in the format they prefer best guaranteeing

system openness to external managers and integration of factory data to the highest production hierarchy levels.

The Data Loggers are easy and simple to use to solve any real-time data storing requirements where all the integrated project tools can be used for obtaining recorded data for viewing, displaying, analysing, formatting, presentation or printing out production reports.



Note that the "Recipes" functionality from the Data Loggers & Recipes resource is always enabled in any purchased product license type. However, the "Data Logger" is optional and therefore not included in the ordinary license version.



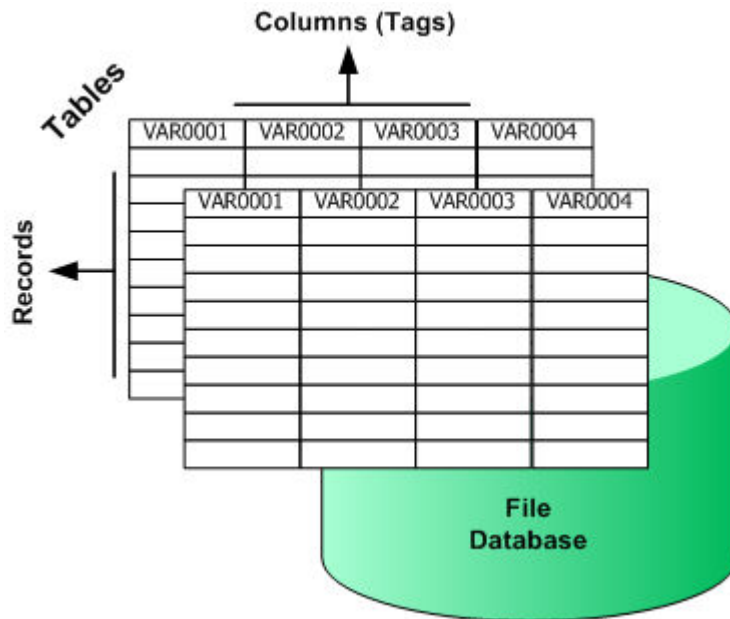
**CAUTION:** Data Loggers based on IMDB do not support Reports from Crystal Reports otherwise use the historicals in ODBC if necessary.

## The DataBase

The Data Loggers and Recipes record data on log files known as a database, whether ODBC or IMDB. These Databases are created automatically by Movicon according to the description in the Historical Log Management section.

However, it is necessary to first configure these Databases' characteristics, which are nothing more but properties of the same Data Loggers or Recipes, being the Database name, the Table name, the quantity of data to be recorded (records) and columns. These are the minimum requirements the Database must know before being created by Movicon. Databases contain data in tables (A linked ODBC Database file may contain diverse tables), which are organized in records, being the line and column numbers representing the variables.

With the IMBD, each column is built from a XML text file.



Organization of data in a normal database

Every time the recording command is evoked, the Movicon Data Logger records in the Database records the variables associated to the columns in order of progression. When the set maximum recorded time has elapsed the recording will recycle on the oldest values.

## Log File Sizes

---

The size of the files is set in the project's properties for the Historical Log, in the Project Historical Log Settings item and is set in the respective Data Logger Recording Time Properties or those of the DB Trace objects. The file sizes are expressed in days with 180 days preset for default. This value can be set as pleased.

The file sizes therefore will depend on the data format being used and on the number of events to be recorded by the system in prefixed time periods.

## DataBase Structure

---

Data Loggers or Recipes are presented with tree structure in the **"Data Loggers & Recipe"** source found in the 'Project Explorer' window. By using this structure you can view the Data Logger or Recipe components which will represent the Database columns.

Each element of the structure which composes the Data Logger or Recipe can be configured through the **"Properties Window"**.

The Data Logger or Recipe structure displayed graphically indicates the Database's components, which can be edited not only in the property but also in the structure itself. The represented order of the columns can be changed as pleased by using the Window's drag functions. For example, select a Data Logger item column) and drag it towards the top or bottom according to where you wish to place it.

In the Database, which is created by Movicon, some system columns will always be inserted by Movicon in addition to the data columns defined by the user. These columns, whose name can be customized through the "Data Loggers and Recipes Common Database Settings Properties" are as follows:

- **Time Column**
- **Local Time Column**
- **MSec Column**
- **User Column**
- **Reason Column**



**WARNING! If a Data Logger or a Recipe use the InMemoryDB (IMDB) for data recording, the table can not include more than 127 columns. Otherwise the Data Logger or the Recipe will not be initialized at the project startup and a message will appear in the log, asking to reduce the columns number. The IMDB has a maximum limit of 127 columns.**

## Database Link to Report Files

---

Even though Reports can be managed according to the different programming techniques, as described in the respective sections, the Data Loggers and Recipe Databases are predisposed to connect to **Crystal Report™** files, according to the **.RPT** format.

Crystal Report™ is one of the most powerful tools for representing data available on Database. For this Movicon integrates the Crystal Report™ interpreted format motor internally, so that a .RPT file can be generated and connected to any Database object, thus permitting you to preview or automatically print the report according to what has been predefined in the .RPT file.

The .RPT files associations to Data Loggers or Recipes and variable definitions for automatic print on event, are done through the properties "Data Loggers and Recipes Common General Properties". The print or view report commands can be executed from any of the project's controls or resources.



**CAUTION:** Data Loggers based on IMDB do not support the Reports from Crystal Reports but are supported by Data Loggers based on ODBC.

## Displaying Database Data

---

Although the displaying of data of Databases linked to Data Loggers or Recipe can be done with different programming techniques, Movicon has a purpose built object for doing this which is available in the Advanced Shapes library and can be inserted into the project's Screens.

The **"DataLoggers/Recipes Window"** permits you to display components inside the Screen which are linked to the Database created by the Data Loggers and Recipes (appear as windows containing tables).

The **"DataLoggers/Recipes Window"** permits you to represent data contained in the Databases in table format, showing the Database columns and records. The object provides a few simple functions which are the Default Query or Dynamic Query, allowing the operator to carry out operating maneuvers in the window such as putting into order, filtering by customizing the SQL queries for DBMS Database.

The **"DataLoggers/Recipes Window"** is described in the respective section.

## Data Loggers as Templates

---

Movicon has a extremely useful function when needing to set many repetitive Data Loggers in the project, which record and sample series of different variables in the same way.

- **A practical example would be:** we have to get our project to manage 50 data recordings with the same sampling time and recording modes. The RealTime Database has declared the 50 variables exchanged with the field and relating to the process information. These variables must generate 50 different columns, one for each piece of data, in a database.

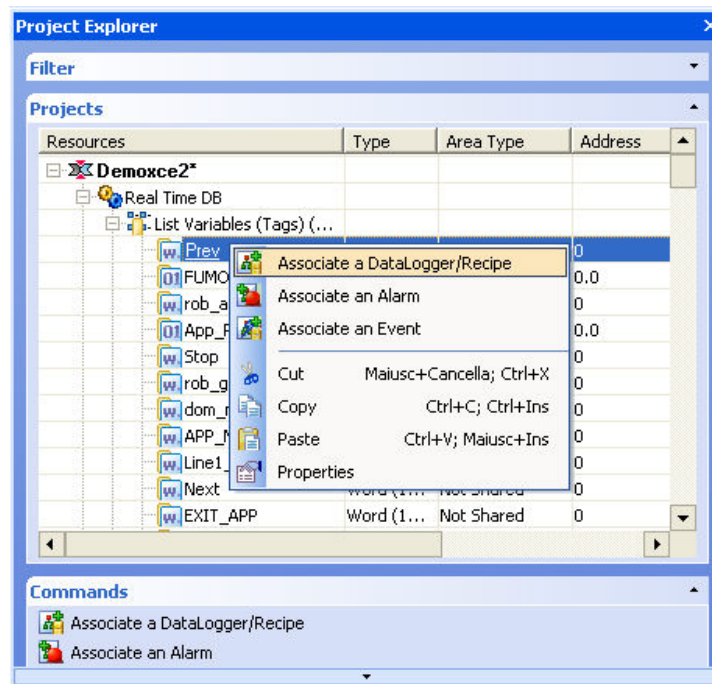
In a case such as this, we need a tool which to quicken the process of assigning Tags to the Data Logger, one which provides you with the possibility to set up the data logger **only once** and associate it to the 50 variables, of the example above, all at once.

A tool such as this exists in Movicon and is called **"Data Logger Template"**.

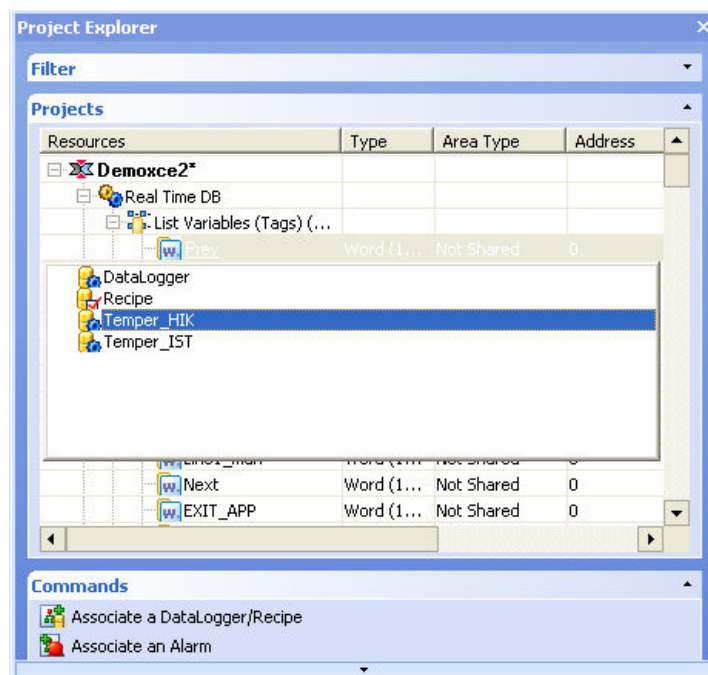
To get this function you have to set the Data Logger as "Template" type in the Data Loggers Resource with the recording mode desired. Then you can select the variables needed from the Variables List from the RealTime Database and use the "Associate a Data Logger" command with the right mouse key and select the Datalogger desired from the window that pops up. The variables (in our example this would be the 50 process variables) will then be linked to the Data Logger Template. Technically Movicon has set 50 different variables linked to the Datalogger in its project on receiving just one click. of the mouse.

This procedure is very handy to have during application development mode when variables that have to be recorded in DataLoggers/Recipes are added to the project at different times.

Data Loggers can be associated to Variables by clicking the right mouse key on the Variable in the "Variable List" group found in the "Project Explorer" window and by selecting the "Associate a DataLogger/Recipe" command, or by using the same command which can also be found in the "Commands" window at the bottom of the "Project Explorer" window.



The request to associate a Data Logger/Recipe will open a another window containing a list of Data Loggers/Recipes available, which should have been inserted beforehand:



DataLogger/Recipe associations to variables can also be done with objects already containing variables to be recorded. Furthermore, if the DataBase already contains data, it will be stored and simply inserted in a new column within the DataBase structure.



**CAUTION:** managing DataLoggers/Recipes as Templates can also be done with the Recipes, but variable associations can only be handled with "Working Variables" and not "Temporary Variables". This means that a Recipe can be used as a Template only in cases where it is necessary to use working variables only and not temporary variables.

## Data Loggers and Recipes Editor

---

Movicon has a powerful tool to configure registration engines managed by projects. The **"Data Loggers & Recipes"** resource is the main tool for generating data files in standard format which can be used for managing reports, recipes, graphics or any other viewing functions, printing or analysing filed data.

The advantage offered by the Data Loggers and Recipes is that each single Database is considered as a an object, completely configurable through the **"Properties Window"**. The property of each single Data Logger or Recipe consents complete customization, whether in recording or visual display unit connecting.

The Data Logger and Recipe objects, after been configured, can be saved in the Symbols library, as Templates. Each single Data Logger or Recipe can be saved in the library individually or associated to a symbol as desired.

This technology, defined **Power Template**, permits the programmer to drastically reduce project developing time allowing the automatic insertion of variables and other correlated Templates, if any, to the symbol being inserted.

The **"Data Loggers & Recipes"** tool is a project resource which can be accessed from the Movicon **"Project Explorer"** window.

## Inserting Data Loggers and Recipes

---

The insertion of Database objects in the Movicon **"Data Loggers & Recipe"** resource is done through the **"Project Explorer"** window. This resource, even when empty, is always present in any Movicon project application.

The Data Logger and Recipe objects can be inserted, modified, deleted and associated with data by using the usual Windows' commands and the Movicon "Properties Windows". The name, number of columns and records and recording mode of the Data Logger or Recipe object are all defined in the "Property Window".

The insertion of a new Data Logger object or Recipe is done in the 'Data Loggers & Recipe' resource from the 'Project Explorer' window. The recording mode and the Data Logger or Recipe data are specified through the 'Property Window'. Each Data Logger or Recipe object has a tree structure, whose branches are a distribution of Database columns.

To insert a new Data Logger or Recipe object you need to select the **'Data Loggers & Recipe'** resource with the mouse and then execute **'New Data Logger'** or **'New Recipe'** from the **'Command'** selection of the **'Project Explorer'** window. These same commands are also made available by clicking the right mouse button on the **'Data Loggers & Recipes'** resource.

New Data Loggers or Recipes can be entered by copying the data from the resource of the same or other projects. In order to do this select the Data Logger or Recipe objects you require for copying, then use the Windows' Copy and Paste standard functions (CTRL+C and CTRL+V command keys, mouse right button).

To delete one or more Data Loggers or Recipes from the project, select them first then activate the 'DELETE' button or 'CANC' on the keyboard or use the mouse right button.

Data Logger or Recipe settings or modification are done through the Movicon 'Properties Window'.

Every Data Logger or Recipe entered will be correlated with one or more columns relating to the data to be recorded. The 'Properties Window' allows the recording mode for the columns to be configured, as illustrated further along

*Note: To carry out the Multi Selection of Data Loggers or Recipes, proceed according to the Windows stands by using the mouse together with the SHIFT or CTRL keys, or the SHIFT key together with the UP and DOWN arrows.*

## Data Loggers and Recipe ODBC Link

---

Apart from setting the ODBC with the chosen format you also have to create a link between the application Server, the table and destination file, in which values will be written by Movicon through the ODBC driver, so that the Data Logger or Recipe can record data using the **ODBC** standard.

It is through this link with the ODBC driver that the Movicon **"Data Loggers & Recipes"** resource is able to record data on file.

The ODBC links have to be registered in the Operating System by using the appropriate ODBC system settings found in the "Data Source ODBC" item from the "Administrative Tools" group on the Control Panel. Movicon also has a guide tool (Wizard) to create ODBC links for all the resources and functionalities that allow its use.

Movicon creates Database files in Access format for default in which all the project's Data Loggers and Recipes will be inserted. Each Data Logger or Recipe is identified by a table within the Database.

The Database file created by Movicon will be inserted in the project's **'DLOGGERS'** folder with the project's name plus the "\_DLR" suffix. For instance, if the project is called 'Project1' the Database file name will then be **"Project1\_DLR.mdb"**.



The ODBC link can be customized, by defining the position, the name and Database file format to be used, by using the "ODBC\_DSN" and "User ODBC DSN" properties which can be accessed from the **"Data Loggers and Recipes Common Database Settings Properties"** of each Data Logger or Recipe object.

Each Data Logger or Recipe object will be represented within the Database by a Table with the same name or with a name defined by the programmer by using the appropriate **"Table Name"** property accessed through the **"Data Loggers and Recipes common General Properties"**.

## Importing Data Loggers and Recipes from other Recipes

---

Movicon allows one or more Data Loggers or Recipes to be copied or moved from one project to another. In order to do this just open the two projects at the same time, so that both are displayed in the 'Project Explorer' window. After having done this one or more Data Loggers or Recipes, or only a few individual columns can be copied or moved from one project to the other by using Window's Copy, Paste or Dragging standard techniques.



The 'Data Loggers & Recipes' resource is saved by Movicon in the "NameProject.movdlrec" file inside the project folder in XML format. The "NameProject.movdlrec" file can also be copied directly when having to copy all the 'Data Loggers & Recipes' resource from one project to another.



# Recipes on Database

---

Often production plants need to record parameters and set-points of a certain product type on file to be kept in archives so that it can be later edited in order to activate the production of a different product. The previous product can be taken from the archives and reactivated in the plant according to the production requirements.

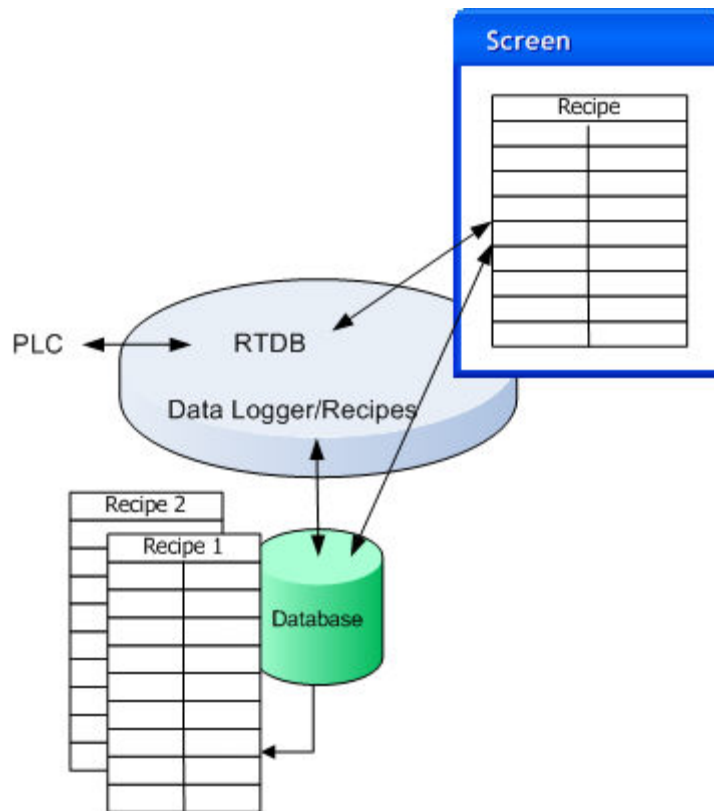
This is usually known as 'production recipe management'. For instance, in a production process where the final product is the result of a mixture of different components specified in percentages, it is evidently necessary to establish which components these are and their percentages in the production recipe to then be activated in function with the final product type you require.

This concept can be applied whenever it necessary to insert, archive, print and activate data whenever required.

Movicon provides you with a '**Data Loggers & Recipes**' resource, a powerful tool for manipulating databases and using them for managing recipes by means of the purpose built functions in the Recipe properties or by means of the Basic Script function.



Movicon can use ODBC or IMDB to record Recipe data on log files. Consult the appropriate chapters on Historical Log Management to evaluate and choose the file format type which suits you best.



A block diagram of how a recipe works: The variables are linked to a Recipe object from the DataLoggers/Recipe resource, which are set to record data in function with the operating commands, which also allows the handling of files.

## How Recipe objects work

---

Even though the Recipe objects are in the same resource as the Data Loggers and share the same functional technology, they have a different way of working. While the Data

Logger records data, at preset times or on event, on files with data recycle management, different operating commands are needed as explained below.

Recipe DB table structure:

Index	VAR0001	VAR0002	VAR0003
Recipe 1	234	234	12
Recipe 2	123	450	34
Recipe 3	321	230	44
Recipe 4	085	450	21
Recipe ...	061	450	13

This table shows an example of data organized in a recipe log file. The table columns are associated to the process variables. The column indicating the recipe index is obligatory.

The techniques used for creating project recipe objects are explained in the next section, Creating Recipes.

However, the basic concept is:

1. The Recipe object, in the Data Logger & Recipe Resource, is defined with its working criteria in its properties
2. The Recipe object's components are the index and Process Variables which are represented by the columns in the table
3. The operating commands in the recipe are "Save", "Activate" and "Delete" as well as the selecting the recipe on file by means of its index. These operating commands become executable by associating the variables to the respective commands in the recipe properties
4. The last operation to carry out involves creating the recipe's graphical interface which can also be done by using the "Create Recipe Editor" command. Movicon will automatically create a new screen containing all the controls needed for managing the recipe: Index Selection (Combo Box), Recipe Variables (Display), Save, Activate and Delete commands (buttons)

## SQL Commands for Recipes in the Database

The Recipes memorized with **ODBC** connection can be done selecting the Database chosen format, with the advantage of being viewed and accessible from other applications supporting the same chosen database format.

The SQL standard language commands, supported by all relational databases, must be used so that Movicon can manage and manipulate the recipes according to the database's technology.

The commands needed to: UPDATE, DELETE and INSERT; their syntaxes below are:

```
UPDATE NameTable
SET NameColumn1 = {value1 | NULL}
[NameColumn2 = {value2 | NULL}]
[WHERE condition-filter]

DELETE FROM NameTable [WHERE condition-filter]

INSERT INTO NameTable [(NameColumn1 [,NameColumn2]...)]
{query | VALUES (value1 [,value2]...)}
```

In order to use these SQL commands in the project's recipe management you have to use the appropriate variables predisposed for this purpose, described in the "**Recipe Execution Properties**" which can edit the query (default query, filter or order) for extracting data from the Recipe Database.

Below are SQL query examples for the three commands seen above:

### Data Update for existing recipe

```
UPDATE TRecipe SET Value01 = 10, Value02 = 20, Value03 = 30, Value04 = 40, Value05  
= 50  
WHERE IdRecipe = 'Recipe001'
```

Where:

TRecipe = name of the Recipe Table contained in the Database

Value01, 02, etc. = name of table column, corresponding to the Recipe's data

IdRecipe = column name 'Recipe Index'

#### Delete Recipe Data

```
DELETE FROM TRecipe WHERE IdRecipe = 'Recipe001'
```

Where:

TRecipe = Recipe Table name contained in the Database

IdRecipe = column name 'Recipe Index'

#### Insert new recipe

```
INSERT INTO TRecipe (IdRecipe, Value01, Value02, Value03, Value04, Value05)  
VALUES ('Recipe001',1,2,3,4,5)
```

Where:

TRecipe = name of Recipe Table contained in the Database

Value01, 02, etc = Table column name, corresponding to the Recipe's data

IdRecipe = column name 'Recipe Index'

## Creating Recipes

---

By using the "DataLogger/Recipe" resource, Movicon permits you to create Recipe objects in the Database to consent recipe and data archive management based on archive files, exploiting the screen graphics potentialities for user interfacing.

The recipe file management is completely **automatic and object orientated**.

To create a recipe follow the steps described below:

## Setting the Variables

---

The first thing to do is to create the necessary variables, if not already done so, in the project. Apart from the recipe process variables (Set Points), there should also be variables, which are those that determine how the recipe works.

These variables should at the minimum be:

1. Index Variable
2. Variable for the Save command
3. Variable for the Activate command
4. Variable for the Delete command

## Creating the Recipe object

---

After having identified the necessary variables, you will now need to select the project's "Data Logger & Recipe" resource. From here, use the "New Recipe " command to create a new recipe in the project. The object will be added to the resource's tree structure.

After having created the Recipe object use the "New Column" command to add **columns** in correspondence with the process variables (Set Points). You can also drag the variables from the Recipe with the drag & Drop technique.

## Configuring the Recipe object

After having inserted the Recipe object and create its data process structure, you need to configure the recipe's operability. Select the Recipe object and access to the "Recipe Execution Properties". Here you will be able assign the variables for the Save, Activate and Delete commands. There are also other operative command types available and which are described in the section about the Recipe Execution Properties.



The recipe Save, Delete, Move Next, etc., commands can be executed by using the appropriate variables which can be associated to the recipe ("Recipe Execution Properties"), and by using the appropriate commands from the object Command List ("Report Commands"). When using the "Report Commands" you will not need to create any variables.

## Creating Recipe screens

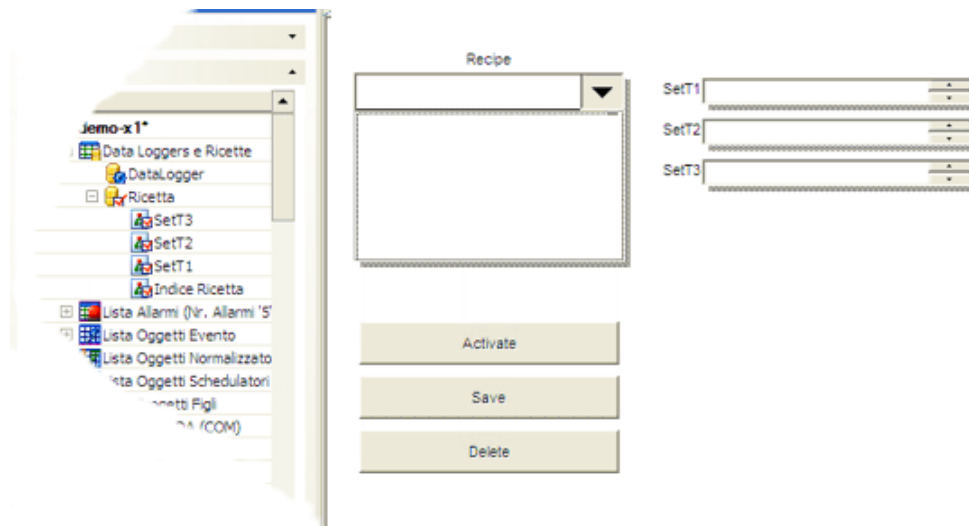
When you have finished configuring your recipe the next and last thing to do is to set its graphic interface which will be the screen and the objects which display and act on the recipe's values.

Movicon has a very simple but powerful command which allows you to **automatically create the recipe screen** with just one click of the mouse. All you have to do is to select the Recipe object and use the "Create Recipe Editor" command. In this case, Movicon will automatically create a screen containing all the graphic controls and objects that the selected recipe needs. All the graphic objects will be already linked to the recipe's variables and the operating commands, leaving you nothing else to do but run the project.



When using this command the recipe's execution variables will be automatically associated to the controls inserted (Save, Activate, Delete buttons etc.) if these have been specified in the "Recipe Execution Properties". On the other hand, if these variables have not been inserted the corresponding "Report Commands" will be associated to the controls instead. Furthermore, if the "List Recipe Variable" has not been inserted, Movicon will create this variable in the RealTimeDB and associate it automatically both in the recipe and in the screen's combo-box object.

The end result of a simple recipe graphic is shown as example below:



This graphic can be completely customized in style and functionality as desired with any other screen graphic object. By using the potentiality of the Symbols and functions from the Movicon Basic Script library you can also insert other graphic components, needed for managing, editing or activating recipes, in ready made screen. Purpose created Recipe Symbols and Resources can be saved as "Templates" in the "Symbol Library", by using the powerful Power Template© technology ready for reuse in other projects without needing to insert lines of codes.

## The Query in Recipes

---

The Recipes are predisposed for extracting data from records registered in the tables by the Movicon registration engine.

Data extraction, defined **Query**, **Ordering** or **Filter**, permit the selecting, filtering or putting the Database contents into order according to the parameters desired, by using the Structured Query Language functions, the standard language of all the databases and is independent from the **ODBC** data format chosen. The Query also permits commands to be executed in the corresponding Database in order to edit or insert its records.

When a Query is executed it creates a **RecordSet**, which is a group of data in memory extracted by the Query which can be used and viewed according to the requirements imposed.

The command query, ordering and filter are set by using the variables that can be inserted in "**Recipe Execution Properties**". If this variables are not declared, Movicon will use the default query, if specified, available in "**Recipe Database Settings Properties**".

Property	SQL Command
Default Query	The default Query, corresponding to a text string in SQL language, supports all the SQL commands but does not create RecordSets. As these commands do not generate RecordSets they can be used for editing or adding records in the database (INSERT or UPDATE are the usual commands).
Default Order	The ordering, corresponding to a text string in SQL language, consents you to set the criteria for putting the extracted fields into order by generating a RecordSet in which the database records are loaded in memory in the order according to the parameters set. Corresponds to the SQL 'ORDER BY' command.
Default Filter	The Filter, corresponding to the text string in SQL language, consents you to define a criteria for extracting data that correspond to specific conditions (ie. Value > 100), by creating a RecordSet according to the criteria selected. Corresponds to the SQL 'WHERE' command.

This functionality's use can be configured through a Recipe property which can be assigned a command, filter or ordering Query, then point to the record from the RecordSet, automatically created by the Query execution, to represent, in the variables associated to the Data Logger or Recipe columns, the values extracted by the Query.

In addition to this Movicon also allows you to use the appropriate Basic Script functions to get complete management of the SQL commands in the queries or Recordsets, making this functionality very powerful and open. Please refer to the section dedicated to the Basic Script for further details on the Basic Script instructions regarding Databases.

## Data Loggers and Recipes common Properties

---

The Data Logger and Recipe objects inserted into the project can be completely customized in the properties. The Data Logger and Recipe objects are built with Tables, Records, recording events, each one structuring the Databases to be generated, and whose properties are accessible from the Movicon 'Properties Window'.

The Data Loggers and Recipes present different operation analogies, some of whose properties are common to both the resources.

## Data Loggers and Recipes common General Properties

---

Each Data Logger or Recipe object, from the moment it is inserted, can be configured in the General properties described as follows. The General properties are used to determine the Data logger's or Recipe's name, their enabling and any associated report file.

To modify the General properties of a Data Logger or Recipe select the object with the mouse and use the Movicon **'Properties Window'**.

### Name

This property allows you to insert the Data Logger's or Recipe's name which has been inserted. The name entered identifies the Data Logger or Recipe object in the **"Data Loggers & Recipes"** resource. If a name is not specified in the **"Table Name"** property, the name of the Data Logger or Recipe will be also used as the table's name.

### Enable Data Logger / Recipe

This property permits you to enable or disable the Data Logger or Recipe. This command consents the programmer to temporary deactivate the recording and running of each single Data Logger or Recipe.



Objects will not get created in runtime mode with the Data Logger or Recipes disabled and therefore will not be accessible through the basic script functions. In addition to this, other viewer objects displayed as Data Logger windows or Trends linked to them will not be capable of retrieving the associated database structure.

### Report File

Each Data Logger or Recipe can be linked to a Report file to display and print documents with formatted texts and graphics. The Report files, containing the print matrix, can be created with **Report Designer** or **Crystal Reports®**. The Report file can be linked to the Database through direct file access or by using the ODBC link created by Movicon.

The Report file can be predisposed to link up to variables recorded in the Database table records.

### Edit Report File

This command starts up the Report Designer for creating or modifying reports report to be associated to the Data Logger.

### Printer name

This property allows you to insert the printer's name where eventual reports associated to Data Logger or Recipe are to be printed. The printer predefined by the system will be used if the printer's name is not specified.

### Table Name

This property allows you to insert the name of the table associated to the Data Logger or Recipe. Each Data Logger or Recipe contains its data in tables composed of records and columns. The table has to have a name in the database because database files linked through the **ODBC** may contain more than one table.

If the table name is not specified, it will be created with the same **"Name"** of the Data Logger or Recipe.

## Data Loggers and Recipes common Execution Properties

---

The common Execution properties of a Data Logger or Recipe allow variables to be defined for the print or reset command executions of the database associated to the Data Logger or the Recipe.

To modify the a Data Logger's Execution property, select the object with the mouse and use the Movicon **'Properties Window'**.

To verify the Execution properties' specifications for Data Loggers refer to the **"Data Logger Execution Properties"** paragraph.

To verify the Execution properties' specifications for Recipes refer to the **"Recipe Execution Properties"** paragraph.

#### **Print Variable**

The 'Print Variable' selection box allows a variable from the Movicon Real Time DB to be associated which will carry out a recorded data print out when it changes to a logical state different from zero. The variable will therefore be set at 'zero' value by Movicon once the operation has been executed. The printer always refers to the Report specified in the Data Logger's "Report File" property.

For further details on Reports management and printing reports please consult the 'Reports' chapter in this document.

#### **Reset Variable**

The 'Reset Variables' selection box allows a variable from the Movicon Real Time DB to be associated, so that when it changes to a logical state different from zero all the values recorded in the table up to that moment will be cancelled to then restart a new set of recordings. The variable will then be set to 'zero' value by Movicon once the operation has been executed.

The next recording command will write data on the first record of an empty table.

### **Data Loggers and Recipes common Database Settings Properties**

---

The Data Logger or Recipe Database Settings property allow you to customize the settings of the database to be created.

To edit the Database settings property of a Data Logger or recipe select the object with the mouse and use the Movicon 'Properties Window'.

To verify the Data base Settings property for Recipes please refer to **"Recipe Database Settings Properties"** paragraph.

#### **Use IMDB Manager**

See paragraph **"IMDB Settings for Recording data"** in the section on **"IMDB"**.

#### **Shared Tables**

See paragraph **"IMDB Settings for Recording data"** in the section on **"IMDB"**.

#### **Save XML File**

See paragraph **"IMDB Settings for Recording data"** in the section on **"IMDB"**.

#### **Crypt File**

See paragraph **"IMDB Settings for Recording data"** in the section on **"IMDB"**.

#### **Write Behind Delay**

See paragraph **"IMDB Settings for Recording data"** in the section on **"IMDB"**.

#### **Max Nr. R cords**

See paragraph **"IMDB Settings for Recording data"** in the section on **"IMDB"**.

#### **Keep the DB Connection open**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

#### **Send Admin.Alerts**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

#### **Max. Error Number**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

#### **Max. Transactions**

See paragraph **"DBMS Settings for Recording data on DataBase"** in the section on **"Historical Log Management"**.

**Max. Cache Size**

See paragraph "**DBMS Settings for Recording data on DataBase**" in the section on "**Historical Log Management**".

**Max. VarChar Precision**

See paragraph "**DBMS Settings for Recording data on DataBase**" in the section on "**Historical Log Management**".

**ODBC DSN**

See paragraph "**DBMS Settings for Recording data on DataBase**" in the section on "**Historical Log Management**".

**ODBC DSN User**

See paragraph "**DBMS Settings for Recording data on DataBase**" in the section on "**Historical Log Management**".

**Time Column**

This setting permits you to insert the name of the Data Logger table's Time Column. The default name will be used if one is not inserted. The Time Column indicates the date and time of the recording in GMT (Greenwich Mean Time) which is universally used as time reference. Time zones are calculated by starting from Greenwich 00 hrs.

**Local Time Column**

This setting allows you to enter the name of the Local Time Column of the Data Logger table. If you do not enter a name here the default name will be used instead. The Local Time Column indicates the data and local time the recording took place.

**MSec Column**

This setting allows you to enter the name of the MSec Column of the Data Logger table. If you do not enter a name here the default name will be used instead. The MSec Column indicates the milliseconds relating to the recording time.

**User Column**

This setting allows you to enter the name of the User Column of the Data Logger table. If you do not enter a name here the default name will be used instead. The User Column indicates the user name which was active when the recording took place.

**Reason Column**

This setting allows you to enter the name of the Reason Column of the Data Logger table. If you do not enter a name here the default name will be used instead. The Reason Column indicates which event triggered the recording (command, change or time).

**Create DB Table**

This command executes the creation of the Data Logger table in the database. If the table is already present the command's execution will cancel the table and recreate it without any data. This means that any data previously recorded will be lost.

## Data Loggers and Recipes common Column General Properties

Each Data Logger or Recipe object has been conceived to manage any required number of values to be recorded which are therefore Database columns. If an ODBC link is used, the maximum number of columns that can be managed depends on the type of Database format being used and the relative **ODBC** links.

Each database column must be associated to a variable from the project through the General properties of a column.

To modify the General properties of a Data Logger or Recipe Column you have to select the object with the mouse and use the Movicon '**Properties Window**'.

To verify the General properties' specifications of a Column for Data Loggers please refer to the "Data Logger Column General Properties".

To verify the General properties' specifications of a Column for a Recipe please refer to the "Recipe column General Properties".



**Name**

This edit box permits you to assign a name to the Database column.  
The name is to be built from a text string which will also be used to identify that same column.

**Variable**

This selection box allows you to associate the required variables, selected among those inserted in the project, to the column.

If an ODBC link is used the data type supported by the column depends on the selected database format and the ODBC link.



When dealing with a **Data Logger**, the variable contents will be the value recorded in the records with the recording command.

When dealing with a **Recipe**, the variable in question will be the one where the recipe data will end up when the activation command is executed.

## Data Logger Properties

---

The Data Logger objects inserted into the project can be completely customized in their properties. The Data Logger objects are formed up of Tables, Records, recording events, some of which form the structure of the Database to be generated and whose properties can be accessed from the Movicon **'Property Window'**.

### Data Logger Style Properties

---

The Style Properties of a Data Logger allow you to define how the data recording modes are to be managed.

Movicon allows data recording to be done on event, on change and on time or with more than one mode at the same time. To modify the Data Logger's Style property, select the object with the mouse and use the Movicon **'Properties Window'**.

**Quality Good Only**

This property allows you to enable the recording of data only when all the variables associated to the Data Logger columns have Good 'Quality' properties. When the setting is left at False the recording will take place independently from variables' quality.

**On Change**

This selection allows you to enable recording on the variable's status change. Recording on status change will not influence the recording on command or on time, as all these modes can co-exist.

Movicon will carry out a recording upon every value change of the variables (columns) associated to the Data Logger object, which will be added to those carried out on command or on time when managed.

Recording will take place each time any one of the variable changed and only in the following conditions:

- the **"Records On Change"** property must be enabled
- if the **"Enable Recording Variable"** has been inserted, the variable must be other than zero
- if the **"Quality Good Only"** property has been enabled, a control is carried out on the quality of the variable associated to the columns which must all have the "Good" quality status
- if the **"Enable Day Timeframe"** property has been enabled, a control is carried out if re-entered within the set timeframe ("Timeframe from" - "Timeframe to")

If the Data Logger's "Recording Variable" has been inserted some considerations have to be added to the recording on change operation.

In this case the operation is as follows:

- Movicon records on the rising edge of the "Enable recording Variable", if a variable associated to a Data Logger column has changed value since the last recording carried out
- Movicon DOES NOT record on the rising edge of the "Enable recording Variable" if all the variables associated to the Data Logger columns have remained unchanged since the last recording carried out. This also goes for all those variables which changed during the period in which the "Enabling recording Variable" remained at zero, but returned back to values last recorded before the "Enable recording Variable" changed to a different value other than zero

#### **Enable Dead Band**

This selection allows the use of a dead band to be enabled for recording on Change. In this case the recording will be executed only when the variable change exceeds the dead band's set value.

#### **Dead Band**

This edit box is used for inserting the dead band value within which the DataLogger will not record on any variable change. This value may be an absolute value or a percentage value according to whether the "Dead Band Percent" has been enabled or not.

#### **Dead Band Percent**

This selection allows the use of the "Dead Band Percent" to be enabled in respect to the variable's value. The percent value will be the one specified in the "Dead Band" property.

#### **On Variable**

This selection allows you to enable record on command, meaning that when a determined variable from the Movicon Real Time DB, defined by the programmer, changes its value from 'zero' to a value 'different from zero'. The variable in question must be selected through the Data Logger's "Recording Variable" property and can be declared any type (bit, byte, word, etc.), as Movicon carries out the recording when the value contained in the variable is different from zero. When the recording has finished, Movicon will force the value in the recording variable to zero.

The data recording sequence is therefore as follows:

1. Set the recording variable to  $>0$  value when you want to carry out a recording
2. Movicon buffers the data in its cache memory
3. Movicon sets the recording variable's value = 0
4. Movicon notifies the DBMS of the data in the cache, while it is ready for a new recording

Recording on command does not influence recording on change or on time, as all modes can co-exist at the same time.

Recordings are only executed when the following conditions have been applied:

- the **"Record On Command"** property must be enabled
- the variable associated to the **"Recording Variable"** property must be other than zero (once recording has finished Movicon will reset to zero)



**Record on Command is done independently from the "Quality Good Only", "Enable Recording Variable" and "Enable Day Timeframe" property settings. Meaning that, independently from these settings, the Data Logger will record each time the "Recording Variable" is set at a value other than zero.**



When data are recorded with ODBC, the recordings are done according to the ODBC modality characteristics. The data sampled from the Data Logger is managed by a cache memory as well as passed over to the Windows ODBC administrator. This may cause a display delay between the sample being effected and the availability of values in the external database file values availability. The ODBC buffer's size can be modified from the operating system's property. Cache unloading can also be forced from the appropriated Basic Script functions.

### On Time

This selection permits you enable the recording cause on a time basis. The recording on time will not influence recording on command or on change, as all modes can co-exist at the same time.

By selecting the recording on time, you will have to specify the time interval within which the recording is to be done by entering the recording frequency, (Hour, Minutes, Seconds and Milliseconds), in the "Data Logger Recording Time Properties".

At each specified time interval Movicon will invoke a recording, which will be added to those done on command or on change whenever they are managed.

Recording takes place at each time interval set only if the following conditions have been applied as well:

- the **"Record On Time"** property must be enabled
- if the **"Enable Recording Variable"** has been inserted, it must be set other than zero
- if the **"Quality Good Only"** had been enabled, the quality of all the variables associated to the columns are controlled to see if they all have the "Good" quality status
- if the **"Enable Day Timeframe"** property has been enabled, a control is carried if re-entered within the timeframe set ("Timeframe from" - "Timeframe to")

## Data Logger Recording Time Properties

---

The Data Logger's Recording Time properties permit you to define the time intervals where the data recordings are to take place when the "Record on Time" item has been selected in the Data Logger's Style.

To modify the Data Logger's Recording Time property, select the object with the mouse and use the Movicon **'Properties Window'**.

### Hour Frequency

This value represents the Data Logger's sampling time in hours. This value is only of significance if the "Record on Time" property has been enabled.

### Minutes Frequency

This value represents the Data Logger's sampling time in minutes. This value is only of significance if the "Record on Time" property has been enabled.

### Seconds Frequency

This value represents the Data Logger's sampling time in seconds. This value is only of significance if the "Record on Time" property has been enabled.

### Milliseconds Frequency

This value represents the Data Logger's sampling time in milliseconds. This value is only of significance if the "Record on Time" property has been enabled.

### Enable Day Timeframe

This setting permits you to enable a data recording day timeframe. When you activate this function, you will need to specify in which day timeframe the data recording is to be carried out (apart from the recording consented by "Recording Variable"). All the Data Logger recording commands evoked outside the established timeframe will be ignored except for the recording executed with the "Record on Command" variable which remains active.



The recording of data in the timeframe, especially if they are variables, can also be executed using "Recording Variable".

### Timeframe from

Setting of the starting time of the Data Logger's data recording when the 'Enable Day Timeframe' is enabled.

**Timeframe to**

Setting of the finishing time of the Data Logger's data recording when the 'Enable Day Timeframe' is enabled.

**Data Max.Age**

This field defines how long the data is to be memorized before being recycled. The time entered here must take into consideration how frequently you intend to carry out data recordings so as to avoid creating tables containing too much data. Practically, more the recording frequencies means more the maximum recording time has to be reduced for each Data Logger.



The maximum recording time to be inserted should be based on your requirements, but you also need to keep in mind both the recording frequency and the type of database to be used. For instance if you use a Access2000 database you will be more limited in the number of data recordings in respect to a SQL Server database.

## Data Logger Execution Properties

---

The Execution properties of a Data Logger allow you to define the variables with which commands will be executed in the Data Logger such as recording on command, enabling Data Logger, etc.

To modify the Execution property of a Data Logger, select the object with the mouse and use the Movicon **'Property Window'**.

To verify the Recipes and Data Loggers common Execution properties refer to the "Data Loggers and Recipes common Execution Properties" paragraph.

**Recording Variable**

In this edit box you can select a variable from the Movicon Real Time DB to be used to execute the Data Logger's record on command. This selection only works if the "Record on Command" property has been enabled. The variable can be declared any type (bit, byte, word, etc.), as Movicon executes the recording when the value contained in the variable is different from zero. When the recording has taken place, Movicon will force the value in the recording variable back to zero.

**Enable Recording Variable**

The 'Enable Recording Variable' selection box allows you to associate a variable from the Movicon Real Time DB which will determine to consent data recording, whether on event or timed. It may sometimes be necessary to impede the recording of values as being insignificant or unnecessary for specific phases of the process.

In this case you can specify a variable which, when the logic state is set different from zero, will consent the recording of data associated to the Data Logger, independently from the fact that the recording be executed on command, upon change or timed.



The recording of data can also be set in timeframes, by enabling the appropriate Data Logger's "Enable Day Timeframe" property.

By leaving the selection empty, as proposed for default, the recording will always be enabled, apart from any time range disabling.

## Data Logger Column General Properties

---

Each Data Logger object had been created to manage the required number of columns. If an ODBC link is used, the maximum number of columns which can be managed depends on the type of database format being use and the relating **ODBC** links.

Each database column must be associated to a project variable through the Column's General property.

To modify the DataLogger column's General property, select the object with the mouse and use the Movicon **'Properties Window'**.

To verify the Recipe and Data Logger common Column General properties please refer the "Data Loggers and Recipes common Column General Properties" paragraph.

#### Type

Through this property you can select which value from the variable must be recorded in the database. The choices are:

- **Instantaneous:** the variable's instantaneous value will be recorded, which is the value contained in the variable at the moment of recording
- **Minimum:** the variable's minimum value obtained in the time interval, that elapses between one recording and the next, will be recorded
- **Maximum:** the variable's maximum value obtained in the time interval, which elapses between one recording and the next, will be recorded
- **Average:** the variable's average value obtained in the time interval which elapses between one recording and the next, will be recorded

**As regards to recording the "minimum", "maximum" and "average" values you will need to take into account that the statistics are reset at each recording, therefore the initial values of these variables will not be considered in the statistic calculations between one recording and the next.**

#### Add Quality

This property, when enabled, allows you to add a column to the Database table which will refer to the variable which reports the Quality Status of the same variable at the moment the recording is executed.

#### Quality Column Name

This property allows you to define a customised name for the Quality column when the "Add Quality" property is enabled. If the field is left empty the default name will be used instead (NomeColumn\_Quality).

#### Add Timestamp

This property, when enabled, allows you to add a column to the Database table which will refer to the variable that reports the date and time of the last variation it underwent.

#### Timestamp Column Name

This property allows you to define a customised name of the Timestamp Column when the "Add Timestamp" property is enabled. If the field is left empty the default name will be used instead (NameColumn\_Timestamp).

#### Add Num.Updates Column

This property, when enabled, allows you to add a column to the Database table which will refer to the variable that reports the number of variations it underwent in the time interval which elapses between one recording and the next.

#### Num.Updates Column Name

This property allows you to define a customized name for the Num.Updates Column when the "Add Num.Updates Column" property is enabled. If the field is left empty the default name will be used instead (NameColumn\_NumUpdates).

# Recipe Properties

---

The Recipe objects inserted into the project can be completely customized in the properties. The Recipe objects are built with Tables, Records, recording events, each one composes the structure of the Database to be generated, and whose properties can be accessed from the Movicon **'Properties Window'**.

## Recipe Execution Properties

---

The Recipe Execution properties allow you to define the variables to execute commands in the same Recipe such as save, activate, cancel, etc.

In order to modify the Recipe's Execution properties select the object with the mouse and use the Movicon **'Properties Window'**.

To verify the Recipes and Data Loggers common Execution properties please refer to the "Data Loggers and Recipes common Execution Properties".

### Execute Variable

In this edit box you can select a variable from the Movicon Real Time DB to use for executing any queries setup for the Recipe. The query to be executed must be contained in the **Query Variable**.

The variable will be then set at 'zero' value by Movicon once the operation has been executed.

It may be necessary to establish the Recipe's data management with an event which will determine the activation of the Query's Selection, Filter and Order, for extracting data contained in the Database according to the search criteria, which can be setup in the properties listed below.

The 'Execute Variable' selection box permits you to select a variable among those contained in the Movicon Variable's Real Time DB to be used for executing any queries setup for the Recipe.

The variable can be declared any type (bit, byte, word, etc.), as Movicon will execute the query when the value contained in the variable is different from zero. After the execution has taken place Movicon will force the execution variable's value to zero.

The query execution sequence therefore results as follows:

1. Set the execution variable to value  $> 0$  the moment the query is needed
2. Movicon launches the query execution in the database and manages the execution status variable being run
3. Movicon sets the execution variable value = 0
4. Movicon notifies the query execution completion by resetting the status variable = 0



In order to execute the query successfully check the status of the query execution being run before executing another.

**When a variable is introduced in the 'Execute Variable' property, the project will load the values of the data file's first record into the variables associated to the columns upon startup of the project in runtime.**

### Move First Variable

The execution of a query, filter or data order determines a **RecordSet**, which is a group of data extracted from the database and kept in the PC's memory. The 'Move First Variable' edit box allows you to select a variable from the Movicon Real Time DB to be used for executing moves to the first record of the eventual **RecordSet** of the selected data.

The rising edge of this variable will permit the extracted values in the **RecordSet** to be represented in the variables associated to the database columns.

For instance, by filtering a data group from the database by executing the Filter command, a **RecordSet** will be generated in the memory containing filtered data. When activating the 'Move First Variable' with a value different from zero, all the extracted values in the RecordSet relating to the first record will be written in the variables associated to the database columns.

The variable will then be reset to 'zero' value by Movicon once the operation has been executed.

#### **Move Last Variable**

The execution of a query, filter or data order determines a **RecordSet**, which is a group of extracted data from the database and kept in the PC's memory. The 'Move Last Variable' edit box allows you to select a variable from the Movicon Real Time DB to be used for executing moves to the last record of the eventual **RecordSet** of selected data.

The rising edge of this variable will permit the extracted values in the **RecordSet** to be represented in the variables associated to the database columns.

For instance, by filtering a data group from the database by executing the Filter command, a **RecordSet** will be generated in the memory containing filtered data. When activating the 'Move Last Variable' with a value different from zero, all the extracted values in the **RecordSet** relating to the last record will be written in the variables associated to the database columns.

The variable will then be reset to 'zero' value by Movicon once the operation has been executed.

#### **Move Prev Variable**

The execution of a query, filter or data order determines a **RecordSet**, which is a group of extracted data from the database and kept in the PC's memory. The 'Move Prev Variable' edit box allows you to select a variable from the Movicon Real Time DB to be used for executing moves to the record before the current one being used in the eventual **RecordSet** of selected data.

The rising edge of this variable will permit the extracted values in the **RecordSet** to be represented in the variables associated to the database Columns.

For instance, by filtering a data group from the database by executing the Filter command, a **RecordSet** will be generated in the memory containing filtered data. When activating the 'Move Last Variable' with a value different from zero, all the extracted values in the **RecordSet** relating to the record before the current one will be written in the variables associated to the database columns.

The variable will then be reset to 'zero' value by Movicon once the operation has been executed.

#### **Move Next Variable**

The execution of a query, filter or data order determines a **RecordSet**, which is a group of extracted data from the database and kept in the PC's memory. The 'Move Next Variable' edit box allows you to select a variable from the Movicon Real Time DB to be used for executing moves to the record after the current one being used in the eventual **RecordSet** of selected data (Each record corresponds to a recipe).

The rising edge of this variable will permit the extracted values in the **RecordSet** to be represented in the variables associated to the database Columns.

For instance, by filtering a data group from the database by executing the Filter command, a **RecordSet** will be generated in the memory containing filtered data. When activating the 'Move Next Variable' with a value different from zero, all the extracted values in the **RecordSet** relating to the record before the current one will be written in the variables associated to the database columns.

The variable will then be reset to 'zero' value by Movicon once the operation has been executed.

#### **Status Variable**

In this edit box you can select a variable from the Movicon Real Time DB where the execution status of any eventual query can be returned.

When a query is to be executed, Movicon permits the logic to be informed about the set query's execution status, by notifying the status through the following bit management on the selected variable:

- Bit 0 = query in Execution
- Bit 1 = BOF
- Bit 2 = EOF
- Bit 3 = Record Cancelled
- Bit 4 = Error

#### **Activate Variable**

In this edit box you can select a variable from the Movicon Real Time DB to be used for executing the activation of the selected recipe. The recipe data temporary loaded into the variable will also be activated in the recipe's own variables with this command.

The variable will then be set to 'zero' value by Movicon once the operation has been executed.

**Save Variable**

In this edit box you can select a Variable from the Movicon Real Time DB to be used for executing the selected recipe save. The recipe data temporary loaded into the variable will also be saved in the Database with this command.

The variable will then be set to ' zero' value by Movicon once the operation has been executed.

**Delete Variable**

In this edit box you can select a Variable from the Movicon Real Time DB to be used for executing the elimination of the selected recipe. The selected recipe will also be eliminated from the Database with this command.

The variable will then be set to ' zero' value by Movicon once the operation has been executed.

**ReQuery Variable**

In this edit box you can select a variable from the Movicon Real Time DB to be used for executing a refresh of the recipe recordset.

**List Recipes Variable**

In this edit box you can select a variable from the Movicon Real Time DB to be used for displaying the list of recipes. When a variable is entered in this field a ComboBox component will be inserted instead of a simple Display in the Recipe Index field when the synoptic is created with the Create Recipe Editor command.

**Filter Variable**

In this edit box you can select a variable from the Movicon Real Time DB to be used for executing a filter of the recipe data (WHERE clause).

**Sort Variable**

In this edit box you can select a variable from the Movicon Real Time DB to be used for sorting out the recipe data (ORDER BY clause).

**Query Variable**

In this edit box you can select a variable from the Movicon Real Time DB to be used for executing a query in **SQL** standard language in data to be selected from the Database.

---

## Recipe Database Settings Properties

The Recipe's Database settings properties can be used to customize the settings of the database to be generated.

To modify the Recipe Database Settings property, select the object with the mouse and use the Movicon **'Properties Window'**.

To verify the Recipes and Data Loggers common Database Settings Properties please refer to the "Data Loggers and Recipes Common Database Settings Properties" paragraph.

**Default Filter**

This edit box permits you to enter a text string containing the filter function in SQL standard language on data to be selected from the Database. The filter functions automatically shows the WHERE suffix in a drop-down window, allowing you to simple indicate the filter parameters.



The text string is static and cannot be changed in runtime. When you wish to use a dynamic filter function in the database object, you will need to use the "Query Variable" which can be setup in the "Recipe Execution Properties".

**Default Sort**

This edit box allows you to enter a text string containing the sort function in SQL standard language in data to be selected from the Database. The sort functions automatically drops



down a window with the ORDER BY suffix in it, allowing you to simply indicate the data sort parameters.



The text string is static and cannot be changed in runtime. When you wish to use to dynamic Sort function in the database object, you will need to use the "Query Variable" which can be setup in the "Recipe Execution Properties".

#### **Default Query**

This edit box allows you to enter a text string containing the query in **SQL** standard language in data to be selected from the Database.



The text is static and cannot be changed in runtime. When you wish to use to dynamic query function in the database object, you will need to use the "Query Variable" which can be setup in the "Recipe Execution Properties".

### **Recipe column General Properties**

---

Each Recipe object has been created to manage any required number of values to be recorded and therefore Database columns. If an ODBC link is used, the maximum number of manageable columns depends on the type of database format being used and the relative **ODBC** links.

Each database column has to be associated to a project variable through the column's General property.

To modify a Recipe's Column's general property, select the object with the mouse and use the Movicon **'Properties Window'**.

To verify Recipe and Data Logger common Column General Properties please refer to the "Data Loggers and Recipes common Column General Properties" paragraph.

#### **Recipe Index**

By using this property you can indicate whether the column in question is to have a recipe index, being that which identifies the recipe. Only one recipe index can be defined for each recipe.



**When the column has been enabled as the Recipe Index, the variable inserted in the "Variable" and "Recipe Temp. Variable" fields must be string type. Otherwise, during runtime, the recipe will notify an error in the historical log in the initialization phase, if the "Recipe Index" column does not have a string variable assigned to. In development mode, however, an alarm message is generated indicating an error if a variable that is not string type has been assigned to "Recipe Index" column.**

#### **Recipe Temp. Variable**

The name of the Variable from the Movicon Real Time DB, which will be used as a temporary variable for holding the recipe's data, is specified in this edit box. The values contained in the 'Temp. Variable' are copied into the recipes output variable only when the recipe's activation command is executed.

## 23. Reports

---

***The Movicon Report management can be done with the Report Designer or with Crystal Report™.***

The Reports are tools for displaying or printing filed process data according to preset time ranges. Thanks to Movicon, the Report management is extremely easy to use, guaranteeing openness in obtaining Report printouts exactly adaptable to end user needs. In fact, in addition to accessing historical data recorded by the Data Loggers, the Reports must guarantee visualization and calculated amounts, totals, averages, variances, selecting time ranges, product batches, operators, shift etc criteria.

This can all be achieved by using the free "Report Designer" tool built-in Movicon, giving users the chance to independently create reports relating to the Movicon databases. In addition, Movicon still has "Crystal Report™" report engine compatibility, but this tool is no longer built-in Movicon and needs to be purchased separately in order to create the reports desired.

The report files created with the "Report Designer" have the ".repx" extension, whereas those created with "Crystal Report™" have the ".rpt" extension.

Reports files can be associated to "DataLogger/Recipe" resources, "Historical Log Window" and "TraceDB Window" objects. When a report file is inserted in the "Report File" property of one of these objects, the "Print" command available in the "DataLogger/Recipe window", "Historical Log Window" and "TraceDB

Window" objects will send the report to be printed. If this report file has a ".repx" extension, Movicon will execute this operation using the "Report Designer" libraries, if the file has a ".rpt" extension, Movicon will use the "Crystal Report™" libraries.

The "View" command is only available for DataLoggers and Recipes, accessed through the "Command List - Report-Recipe", and consents the report file to be opened and viewed instead of sending it directly to be printed.

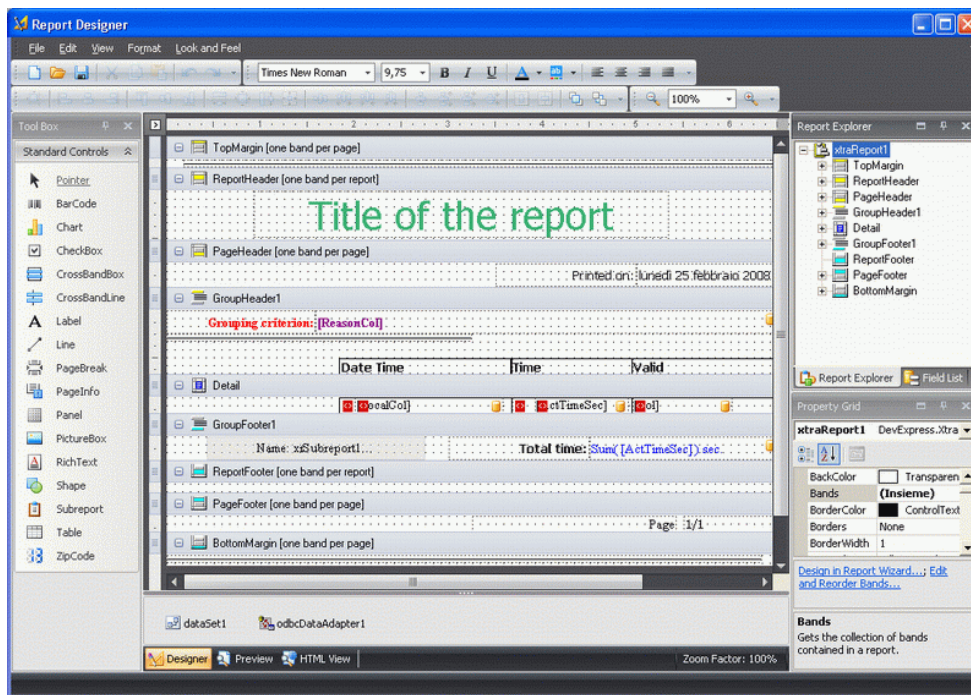
### Report Designer

---

The Report Designer is an extremely powerful tool that has been integrated in Movicon for give users the possibility to automatically create reports relating to the Movicon databases, without having to purchase supplementary software packs. The Report Designer Libraries are therefore installed with Movicon totally free without needing you to enable any license options in order to use them.

The basic steps in creating a minimal report are briefly described below:

The Report Designer can be called through "Edit Report File" command found in the Data Logger or recipe properties of "Historical Log Windows" or "TraceDB Windows". In this case the Report Designer will open in edit mode:



The Report Designer opens showing its "Tool Box" on the left and the "Report Explorer", "Field List" and "Grid Property" on the right. The report Layout is in the centre.

By using the Movicon commands the report will open with the database connection already configured with the original object's data table fields listed in the "Field List" for reference (DataLogger/Recipe table variable TraceDB or Historical Log).

The report is composed of "Band" objects, which are inserted using the "Insert Band" command accessible with a right click on the report layout area. The Bands are:

- **Top Margin:** one per page only. Represents the top limit of the report page.
- **Report Header:** one per report only. The first thing printed and only on the first report page.
- **Page Header:** page header and is printed at the top of every page.
- **Group Header:** you can use more than one group header per report to represent the beginning of a data group.
- **Detail:** contains the data to be shown.
- **Group Footer:** closes a data group and always corresponds to a Group Header.
- **Report Footer:** printed at the end of the report on the last page.
- **Page Footer:** closes each report page.
- **Bottom Margin:** one only per page. Represents the report page's bottom limit.

Only the Detail Band is needed to build a simple report. The fields you wish to display can be dragged from the "Field List" and when the report is executed these fields will show with all the records existing in the table.

#### Filters

You can use the xtraReport object's Grid "FilterString" property (select the xtraReport in the Report Explorer Window). This property must contain the expression criteria (WHERE (Es. [col] > 0, where "col" is the name of a data table column).

#### Groups

You will need to insert a Group Header Band and use the "GroupFields" property for setting the second field to group data. You can also specify the data order in the same property.

#### Data Summary

The report can execute summaries (totals, counts, averages,...) on data groups. This is done by inserting a "Label" band, linked to a field from the Field List. The "Label's"

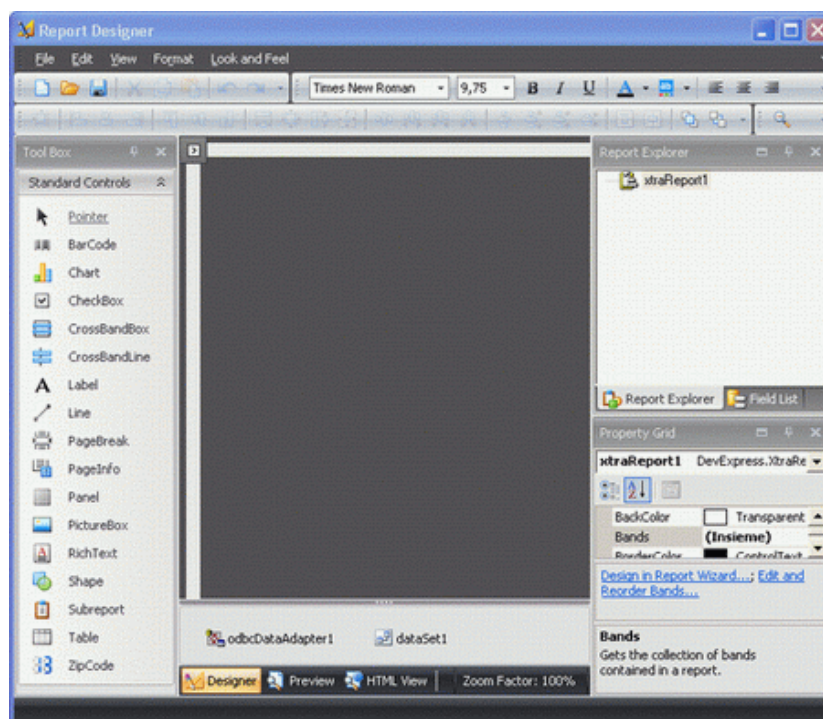
"Summary" property allows you to set the summary type you wish to create using the dialog window which contains:

- **Bound field:** name of field in which operation is to be executed
- **Summary function:** operation type to be executed
- **Format string:** format type for results where you can add measure units as text
- **Ignore NULL values:** specifies whether or not to ignore NULL field value records
- **Summary Running:** defines the environment in which to execute the operation.  
Obtains these values:
  - None = no calculation
  - Group = executed for all group members. This value resets at the beginning of the next group
  - Page = executed for the values presented on one page
  - Report = executed for all the values in the report

## How to create a Report

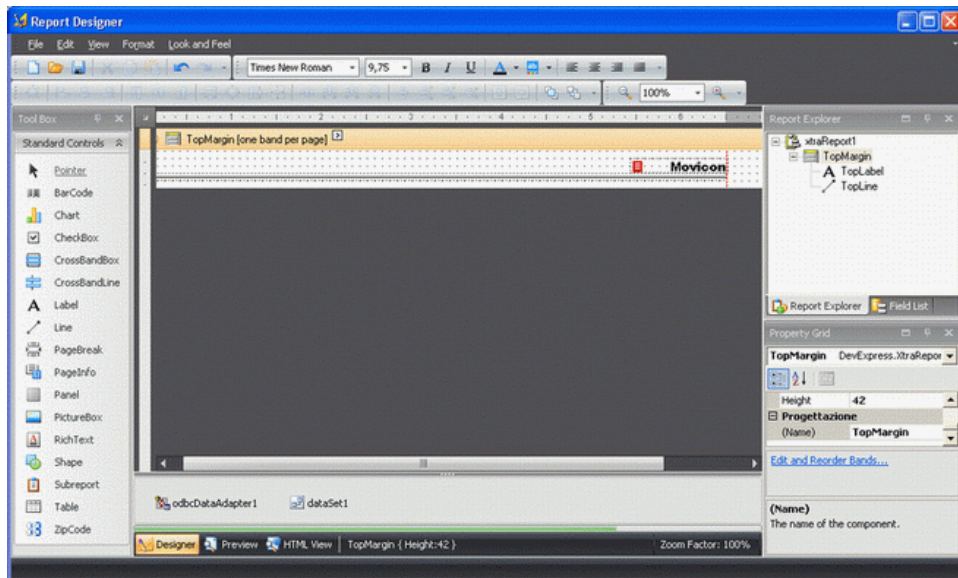
This is an example to show you how to create a report using the Report Designer for displaying data recorded by a Data Logger:

1. First create a new project and insert a Data Logger for recording three temperatures. Set three Float variables and call them "Temp\_A", "Temp\_B" and "Temp\_C". Then add three columns to the Data Logger, set a 1 minute recording time and generate the table using the "Create Table" command from the Data Logger's "Database Options".
2. Now proceed with creating the Report. Click on the "Edit Report File" command from the Data Logger's General properties to open the Report Designer window. The ODBC link to be used will automatically be passed to the Report Designer and therefore the Data Logger table structure with relating fields should be ready and available in the "Field List" window.
3. The window's center pain is the workspace and will appear completely empty. Here you will need to add the different report sections one at a time.

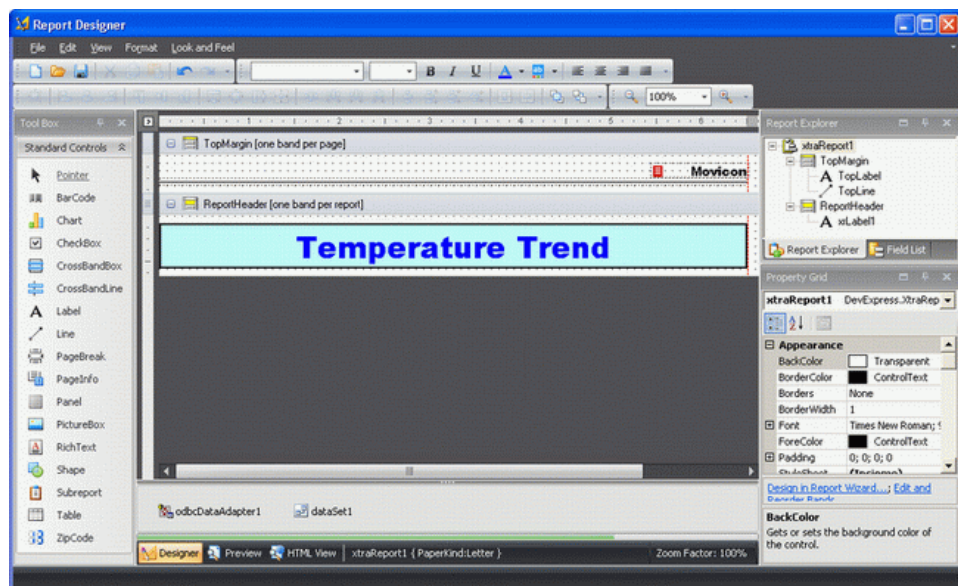




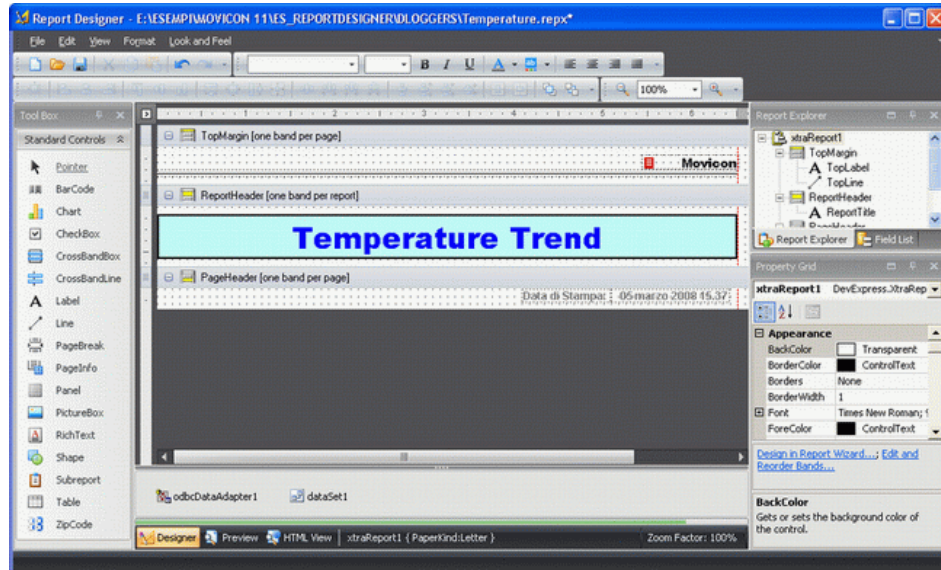
4. First start with inserting the sections, or Bands, to be displayed in the report. Normally the "detail" band is the most important one because this is where the data to be displayed is inserted. To insert a Band right click on the report workspace and select the "Insert Band" item and add the "TopMargin" Band. Only one Band per page can be inserted and represents the pages top limit. Now add a horizontal line and a label, with Movicon written inside, within the Band by selecting and inserting the "Line" object from the "Standards Controls" window located to the right of the editor window and drag it horizontally across Band. Then select and insert the "Label" object in the Band, right click it to open its "Property Grid" window, or click on the arrow which appears on the top right of the control to access its main properties, and insert "Movicon" in its "Text" property.



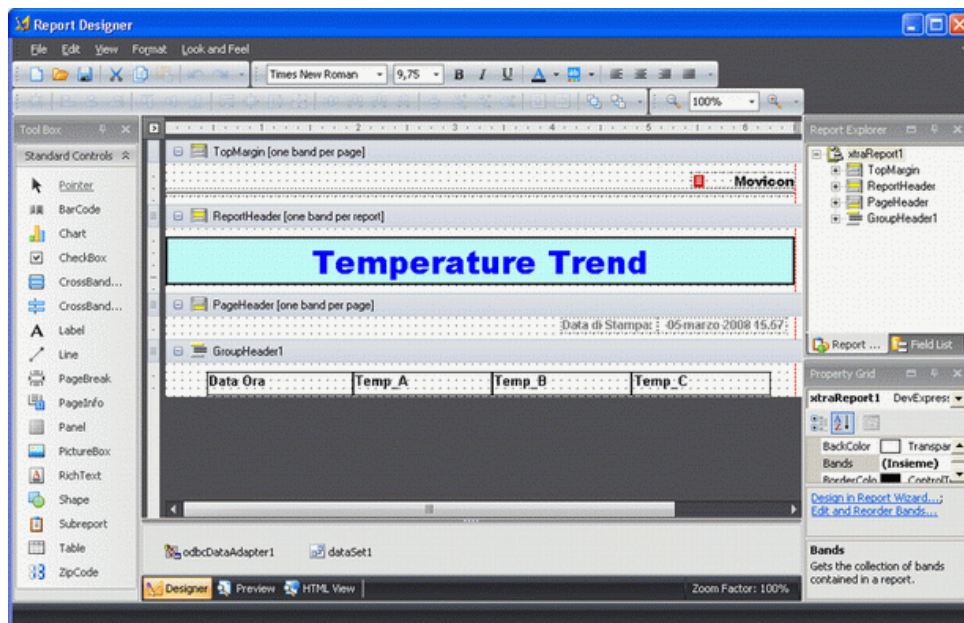
5. Now insert the "ReportHeader" Band. This Band represents the report's heading and will be printed at the beginning of the report on the first page only. To insert the heading select and insert another label into the "ReportHeader" band, size it and through its properties give it a color and insert the "Temperature Trend" heading in the text property. Then apply bold and change the heading font size and center it within the label.



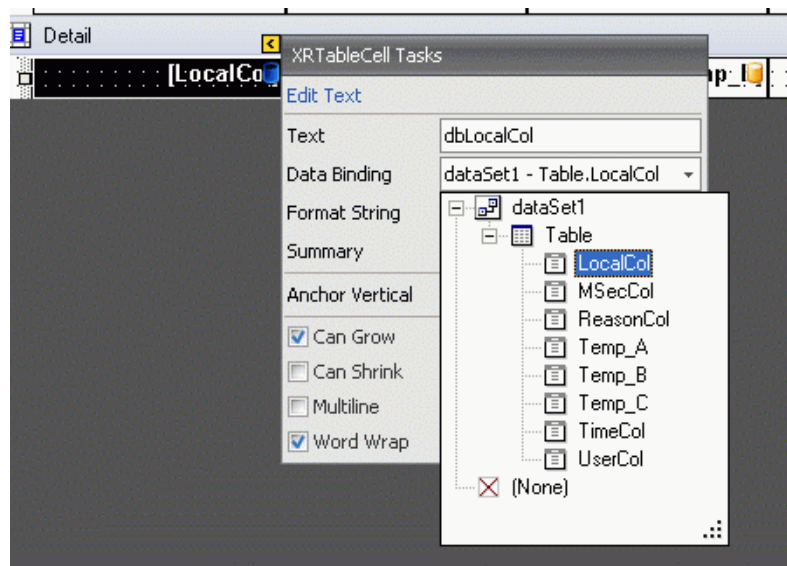
6. Insert the "PageHeader" Band. This Band represents the page heading and is printed at the top of every page. Insert a "Label" in this Pageheader band with text as "Print Date". Then insert an "PageInfo" object with the Label and select the "DateTime" item in its PageInfo property field. After which click on arrow, to the top right, to open the "XRPageInfoTasks" window edit the data and time format through the "Format" field as desired.



7. The aim of this report is to create and display a table with date retrieved from the database. In order to do this, insert the "GroupHeader" Band. This Band is used for showing a group heading and there can be more than one in a report. This group header will represent the beginning of a data group and you will only need to insert one group to achieve the aim of our project. In this band a line is to be inserted with the headings of the table columns. Therefore add a "Table" object from the "Standard Controls" window. The Table object is inserted with three columns for default but you will need four, so select the object and right click to add another column using the "Insert - Column To Left" command. Now size the columns to fit across the page and insert headings to be displayed in each cell's tasks properties done by clicking top right arrow to open each cell's tasks window. The texts for the four columns are "DateTime", "Temp\_A", "Temp\_B" and "Temp\_C".

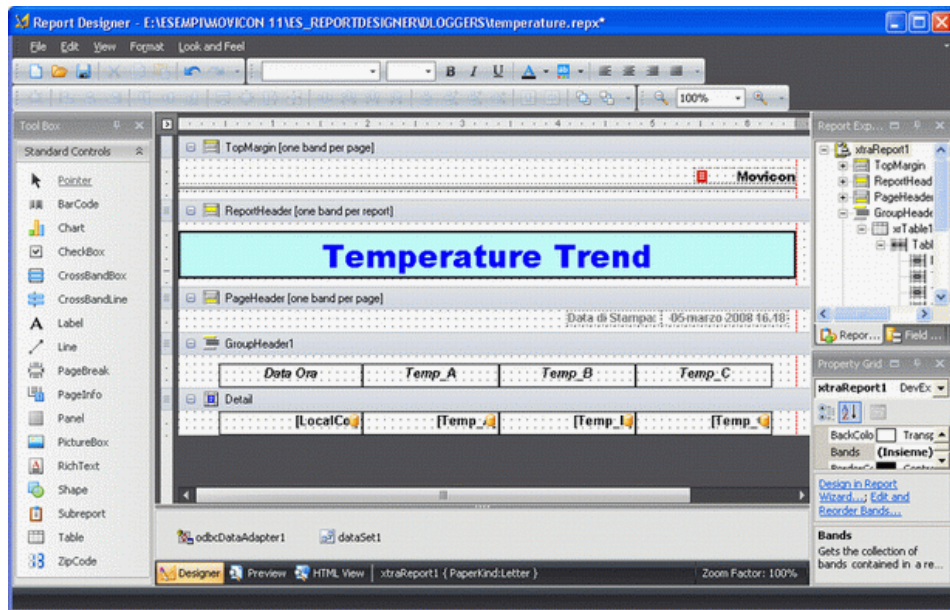


8. Insert the "Detail" Band. This Band will contain the data to be displayed for the data group. Here you must create a table with the same sizes inserted for the "GroupHeader" or copy&paste the table to save time. After you have done this, associate each table cell with the database field from which data is be retrieved. First start by selecting the first cell on the left and open its task window. Then select the "LocalCol" table column from the "Data Binding" field.



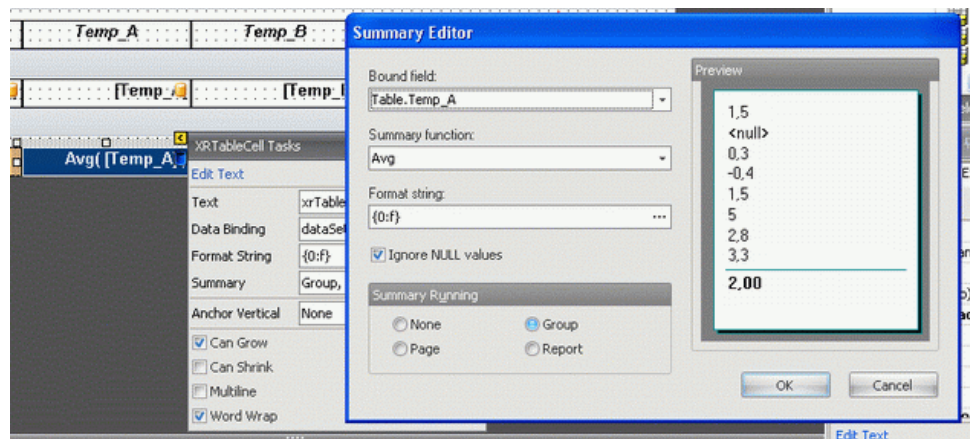
Perform this operation for the other three columns by associating the "Temp\_A", "Temp\_B" and "Temp\_C" field respectively. Also set each cell with a data format where the "LocalCol" column will be in "Date" format and the other remaining three will be "Float" type.





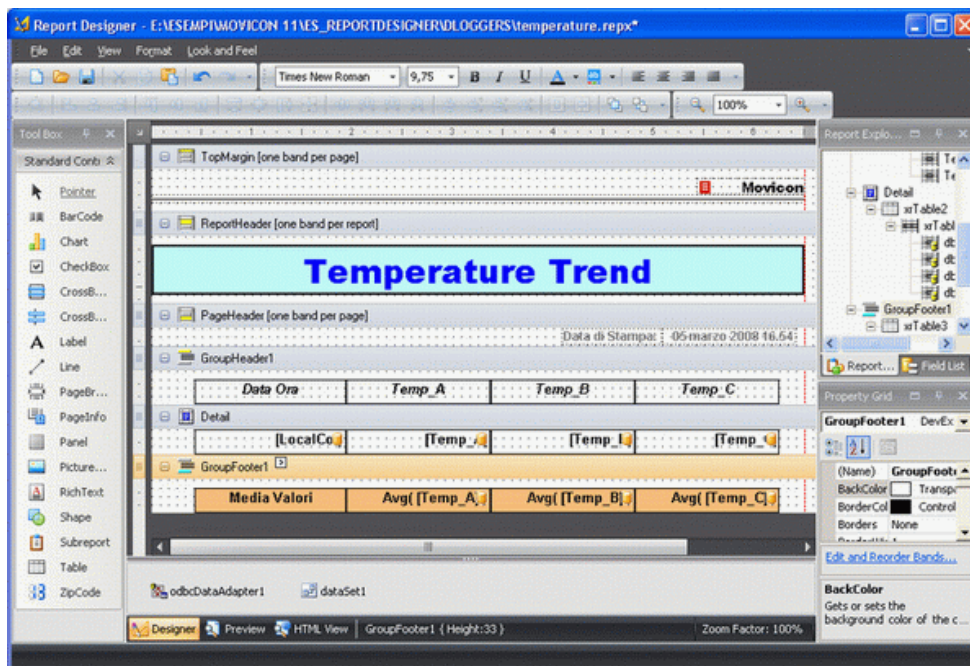
8. Insert the "GroupFooter" Band. This Band closes the group and always corresponds to the "GroupHeader". This Band will show the average values extracted for the three temperature columns. Insert another table with four fields columns as before. The first field just add "Average Values" as text. Select the second field, open its tasks window and click on the "Summary" field to open another window where you can define the formula to apply to the field. The window items must be compiled as follows:

- **Bound field:** select the table field in which to apply the formula, in our case "Temp\_A"
- **Summary function:** select the function type to apply, in our case "Avg"
- **Format string:** define the data format type, in our case "Float"
- **Ignore NULL values:** enable this check box when wishing to ignore null values
- **Summary Running:** select which environment to run operation, in our the case "Group"

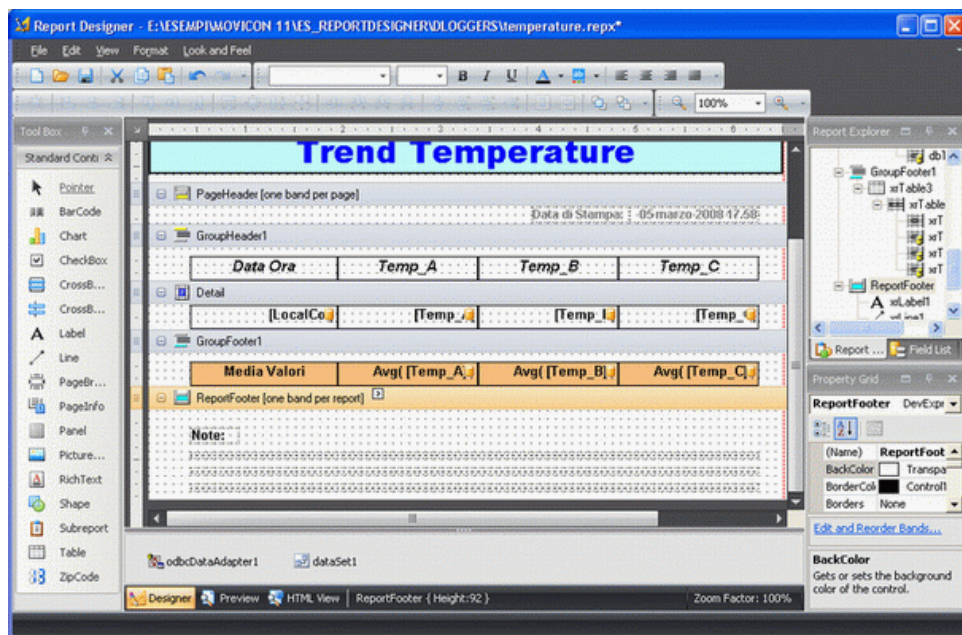


Repeat the same procedure for the other two "Temp\_B" and "Temp\_C" table fields. Apply a background color to average values table.

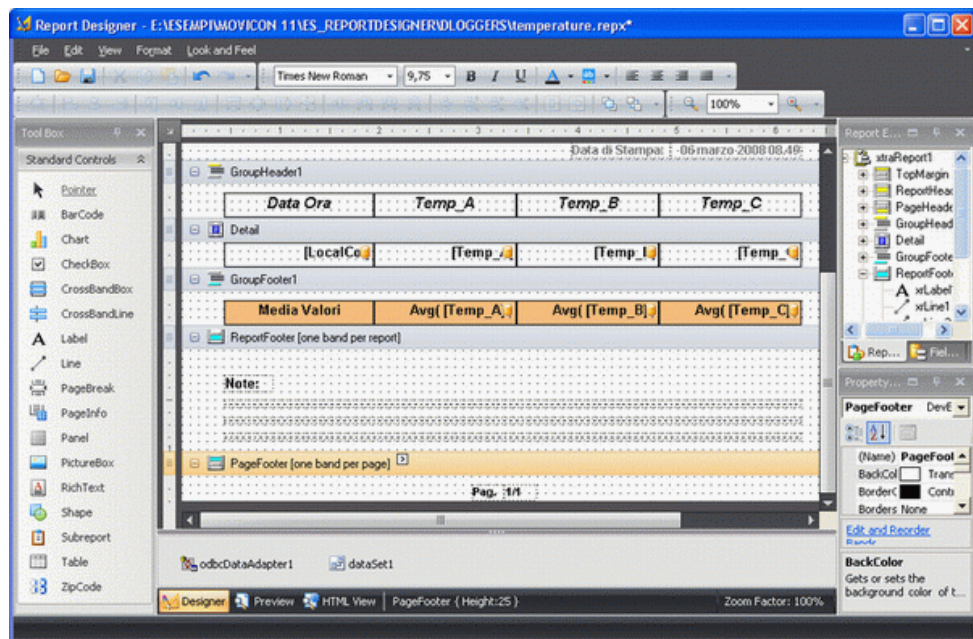




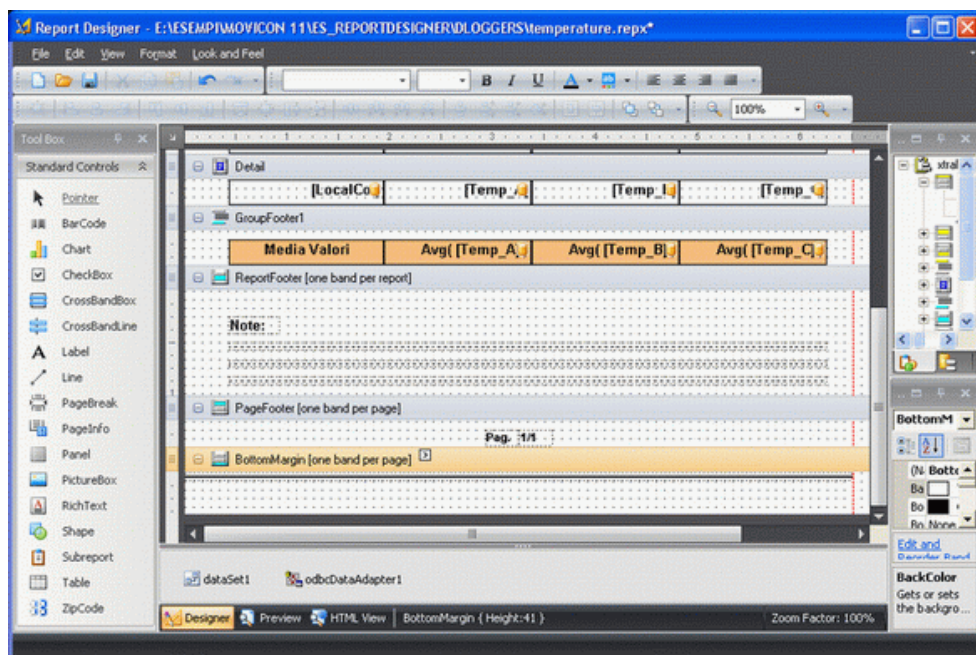
9. Insert the "ReportFooter" Band. This Band will printed at the end of the report at the last page. This Band is will be inserted with a field so that the operator can enter notes. Simply insert a "Label" and three lines and modify them into dotted lines.



10. Insert the "PageFooter" Band. This Band closes each report page. In this Band you are going to insert a field for display the page number. Therefore add a "Label" with the "Pag." text and a "PageInfo" field and select the data type to be displayed, ie. "NumberOfTotal" (displays the current page and the total number of pages).



11. Insert The "BottomMargin" Band. Only one of these a page can be inserted and represents the page's bottom limit. Add a horizontal line in the Band just like the one inserted in the "TopMargin" Band.



12. This ends the report and if you have data already in the database you can see how it will look by clicking on the "Preview" tab on the bottom Report Designer Bar to get a report preview.





# Crystal Report™

---

Crystal Report™ is a very userfriendly and powerful tool for creating basic graphics dedicated to data lists, tables, reports and any other graphical format formats for presenting database data tables created with the Movicon recording tools.

Report pages created with Crystal Report™ will have the ".rpt" extension. These reports must be generated using this tool, purchased separately, and they will be linked to the data table generated by Movicon.

Movicon supports Crystal Report™ version 10.0 or later.

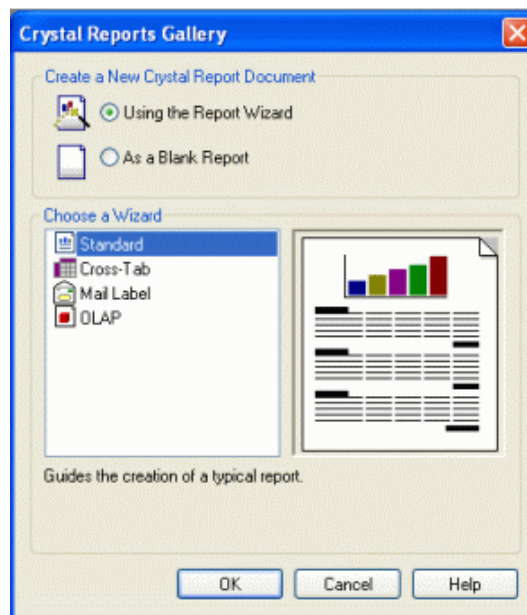
## Creating Reports with Crystal Report

---

The procedures for creating data reports using the integrated Crystal Report with Movicon are described below. In order for these procedures to work properly you will need to use the Crystal Report version 10.0 and subsequent versions.

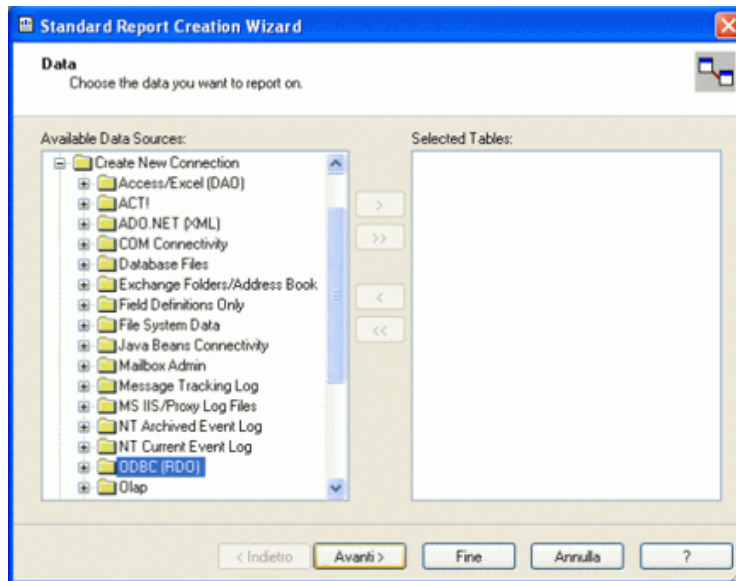
Let's suppose we want to create a project containing a DataLogger which records three Movicon system variables, in order to simulate a random distribution of values recorded in intervals of five seconds between each one. The project must be started up at least once (with ALT-F12 or with the run button) to allow Movicon to create the database.

The next step involves the use of the Crystal Report for creating the report required. When the program is started up a screen is displayed showing a dialog window asking you to choose a report type: select the "Standard" type and click the "OK" button to continue.



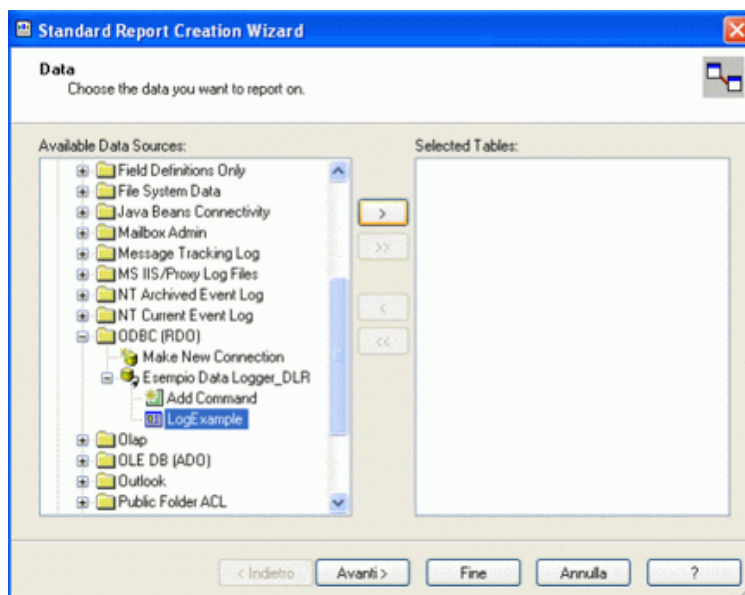
Crystal Report: New Project

Now a list of the types of database connections will be displayed in a tree structure: click to extend the "Create New Connection" node and select "ODBC (RDO)". A list of already existing connections will appear among which you should find the one created previously by the project, with the name defined as "ProjectName\_DLR". By clicking on "Next" a request to enter User Name and Password for the database will appear: leave these fields blank and click on "End".



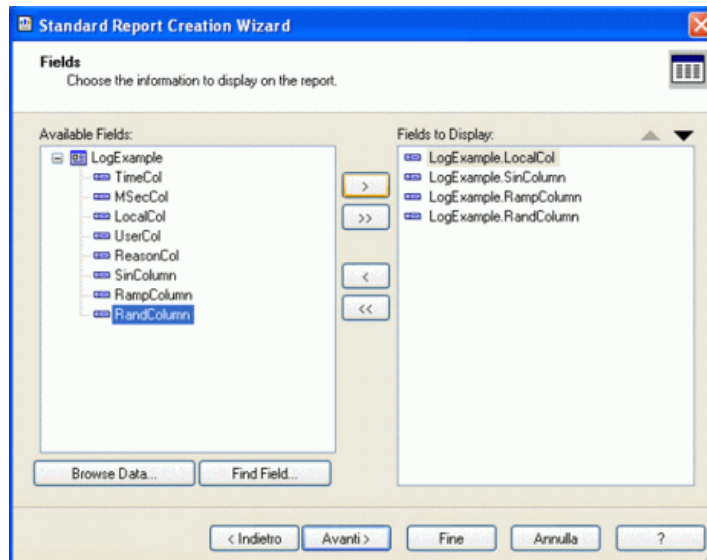
Crystal Report: Create a new ODBC Connection

At this point select the chosen ODBC connection to get the list of its table names (in the example shown the table is called "LogExample" like the DataLogger's name). Click on the button with the arrows pointing towards the right to add the item to the list of selected tables and click on "Next".



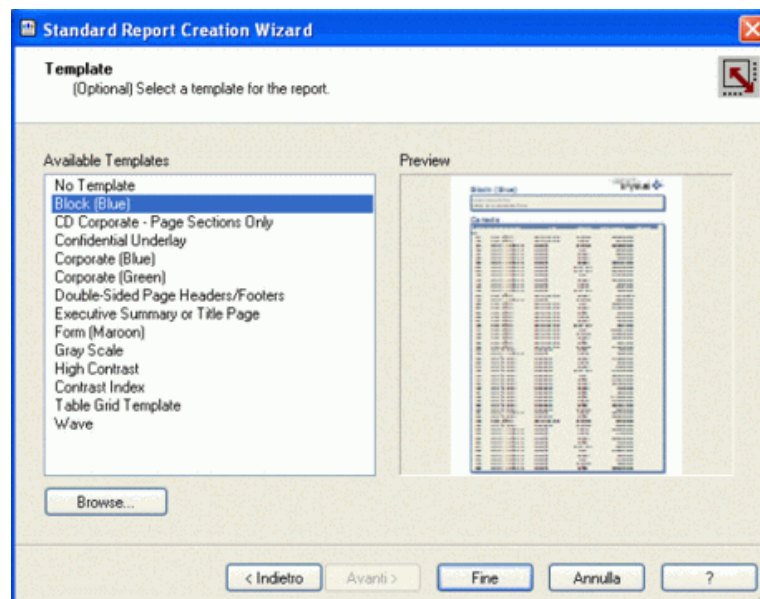
Crystal Report: Add Table

On the next page you will find the fields in the previous selected table listed on the left; double left click on the fields or select them and click on the arrows pointing right to add the fields to the list on the right which contains the database fields which are to be shown on the report.



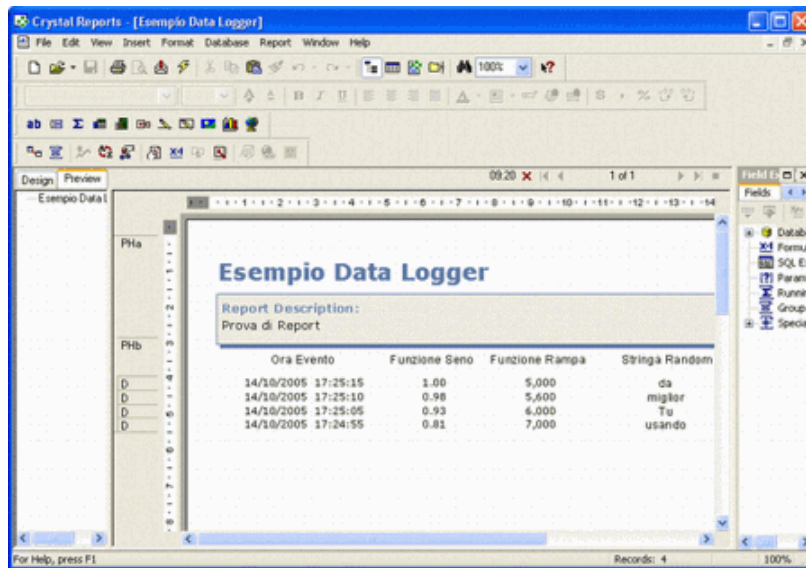
Crystal Report: Add the fields to be displayed

Click on "Next" to group the gathered data into one or more columns and, on the next page, set the data filters. To conclude, choose a template for displaying the data from the last page. If you want to create a simple list without any graphic items just select 'No Template' from the list.



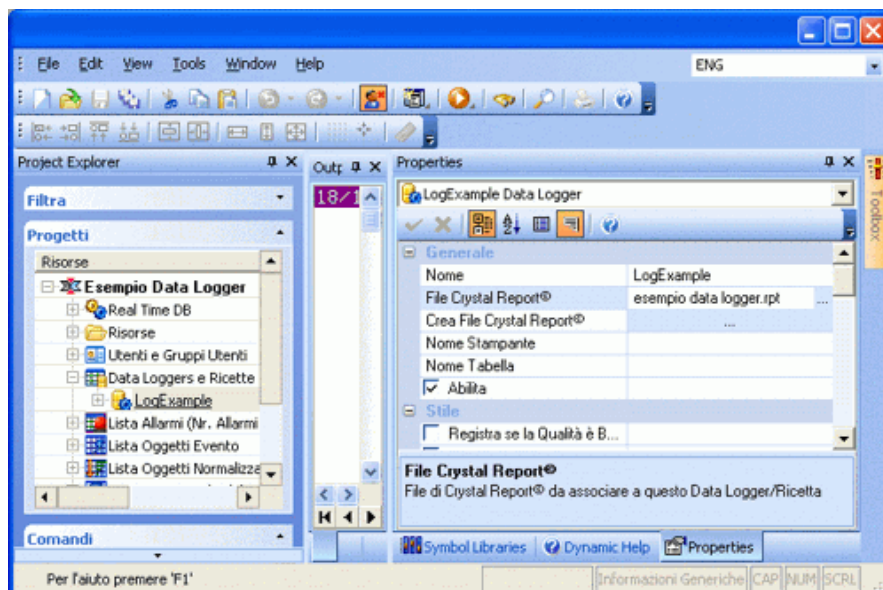
Crystal Report: Choose Template

Click "End" to open the report you have just created using the procedures above. You can adapt the report by using the Crystal Report function to suit your needs by adding information on data or calculations on values like average values, sum of data, ecc. (see Crystal Report documentation for information). Save the report on file to end the report creating procedures.



Crystal Report: End Result

You now need to link the report file to the Movicon Datalogger. Open the project and display the DataLogger's properties window. In the "General" property click on "File Crystal Report" and select the Crystal Report (.rpt) you created previously. Click on "OK" and the name of the file will display in the DataLogger property. By doing this you will be able to display the report directly from the screen during runtime. To do this, add a button to the screen and select the "Report Command" by specifying "View" as its action and the name of the DataLogger as the object. When this button is clicked on during the runtime mode the Movicon Report Viewer should open and display.



Movicon: Chosen Report File

## 24. IL Logic (Instructions List)

---

*The Movicon IL Logic permits the use of PLC type logic tasks in projects for managing the control of variables or I/O.*

The **IL Logic (Instructions List)** is the key feature of the Movicon system. This resource is commonly known as 'SoftLogic' and can has all the effects of a PLC (Programmable Logic Controller) which runs its program in background independently of the system's logic, graphic or historical engines.

The IL Logic offers a number of obvious advantages. Basically it allows you to perform any type of non-standard executions using internal PLC type programs (SoftLogic).



Considering the possibility of installing or connecting digital or analog I/O devices directly to the PC, the Movicon system is capable of replacing the plant control device, such as the PLC (replaced by the IL Logic).

The IL Logic can be associated to projects, where it is known as **General Logic**, or it can be associated to symbols (drawings and controls) and to screens where it is known as **Local Logic**. The fundamental difference between the General Logic and Local Logic are:

1. At project startup the General Logic is run and continues running cyclically until the project stops. The Local Logic, however, is run only when the object containing it is active, ie. when loading a screen in RAM
2. The General Logic run has priority over the Local Logic

The Logic is edited through the **"Logic Explorer"** window which displays the logic associated to the component selected at that moment.

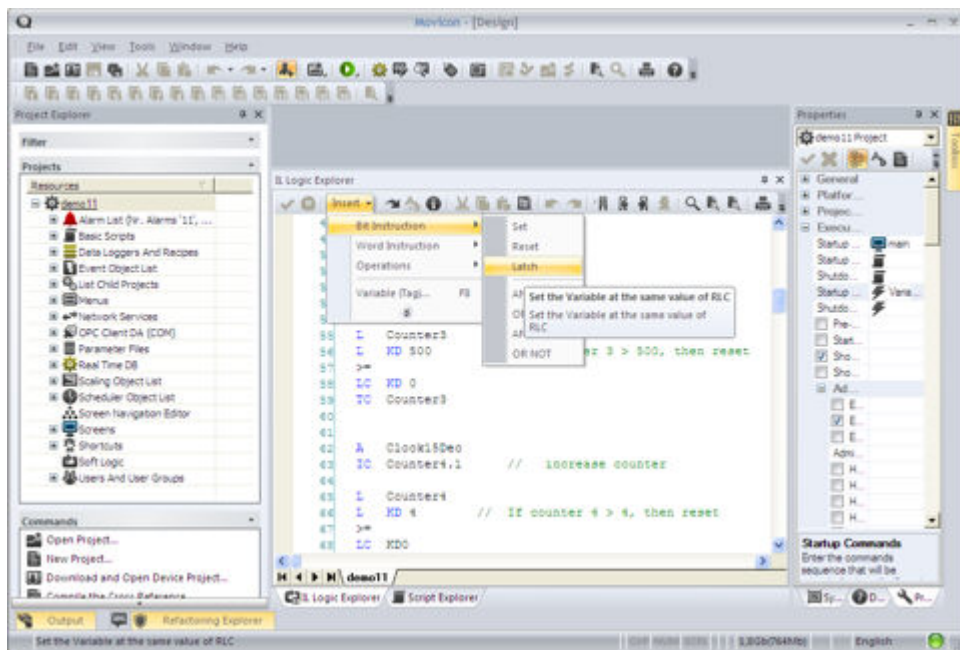
### Inserting IL Logic instructions

---

Inserting instructions which form the Movicon IL Logic Resource is done by using the appropriated editing window called **"Logic Explorer"**, which is accessed from the 'View Menu'.

The **"Logic Explorer"**, illustrated below, allows you to type the instructions and the variables which form the IL Logic program as a normal text editor, permitting the use of graphic formatting to make the program easier to understand.





The Movicon Logic Editor should be used as a normal Windows standard text editor. The programmer, while inserting the program, should naturally respect the programming syntaxes when entering logic instruction for variable combinations.



The programming has been made easier by the possibility to insert all the instruction by using the appropriated commands from the Insert Menu found in the 'Logic Explorer' window. This makes editing easier for the not-so-expert programmer by guiding them through the necessary procedures.

Any syntax errors made during programming unguided by the Insert Menu will be indicated and highlighted in the compilation phase. However, you can use the 'Compile-Check Syntax' command which is available from the 'Logic Explorer' window to check to see if there are any errors.

The opening of the Logic window determines the displaying of the empty editing area with the cursor positioned at the top left. You can then proceed with writing the logic program by respecting the Movicon Editor's programming syntaxes. Program editing works with the ordinary procedures of a normal text editor, apart from the fact that the programming syntaxes are automatically acknowledged and represented with default colors.



You are unlimited with the number of comment texts that can be written as long as they are preceded at the beginning of the line with the '/' character.



Access to the variables is only permitted by using the Variable's name. You cannot access to variables through their absolute addresses (even when the variables are mapped in the Input, Output and Flag shared areas).

The IL logic instructions have also been adapted to the AWL syntax of the Step7. Therefore some instructions may be written in three different syntaxes: Movicon standard syntax, English Step7 syntax, German Step7 syntax. You can use on of these three syntaxes in mixed mode in the same listed IL logic. For further information on the instruction please consult the relevant tables.

## Unsupported Variables

---

Some variable types cannot be used in the Movicon IL Logic especially those listed below:

- **Double Variables (64 bit double precision):** these variables cannot be used in as the ACCU1 and ACCU2 registered in 32 bit
- **String and Byte Array type variables:** these variables cannot be used as their format is not supported in IL Logic
- **Variables with names beginning with "K":** these variables cannot be used because Movicon would expect a constant to be loaded. If there are any "Kf1" variables in the Real Time DB, Movicon would not know whether to load a constant or the variable. Therefore the IL Logic does not support variables which begin with the letter 'K'

## Variable Access

---

You can address variables in the IL Logic which have been declared in a different hierarchical level in respect to that of the object in which the logic has been inserted, such as a Father or Child project RealTimeDB Variables with the same Local screen variable name. You can use the '..\<VariableName>' and '<ChildProjectName>\<VariableName>' suffixes to indicate the variables. When the name of the variable is indicated only, a search is made for the context in which the logic is executed.

# Logic Programming

---

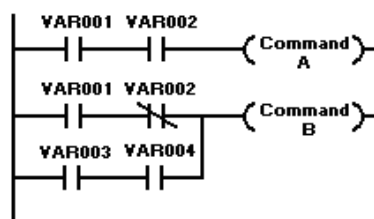
The Movicon Logic Editor provides you with a powerful set of logic, mathematic or compare instruction for combining variable with each other in order to activate logic commands which form the base of the project.

The Movicon programming logic language is in instruction lists (known also as AWL or IL Instruction List).

This type of language, defined by the IEC 1131-3 standards to which Movicon abides by, require a vertical list of logic or compare instructions which, combined with each other, determine a binary result which will be referred to as **"Combined Logic Result CLR"** or **"LCR"** from now onwards.

The Logic Editor can be used indifferently either in the Project's General Logic or in the Local Logic.

**Example of contact logic representation (Ladder)**



**Example of instruction list logic representation (AWL)**

```
AND VAR001
AND VAR002
= Command A
```

```
AND VAR001
AND VAR002
O
AND VAR003
AND VAR004
= Command B
```

## Combined Logic Result CLR

---

The combination of two or more digital variables constitutes the basis of a logic program and always determined a **'Combined Logic Result'**, being a binary logic state that determines whether or not to activate the next operation.

The combined logic result **"CLR"** can assume two logic states only:

**"zero" (0, FALSE, LOW)**  
**"one" (1, TRUE, HIGH)**

The CLR always works on the combination of instructions with variables in bit or the result of comparison operations.

The other operations (mathematical, arithmetical, Load, Transfer) do not effect the CLR.

If for example the logic result of a given sequence of instructions is in logic state '1', the next command will be executed or ignored if the logic state is otherwise.



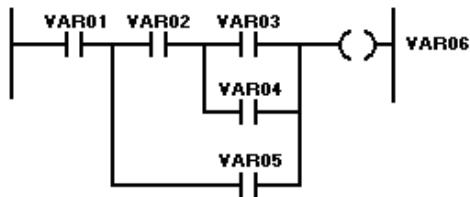
All the mathematic, arithmetical, load and Transfer instructions do not effect the CLR but they can be conditioned by it. All these instructions can be conditioned in the execution of the combined logic result of the previous instructions.

## Bracket levels for the CLR

The Movicon logic programs are capable of handling up to three bracket levels containing logic instructions. Enclosing logic instructions between brackets means that logic conditions can be determined whose CLR result is placed in relation to the previous CLR result. In practice, in an AND combination inside brackets, an AND is placed between the CLR in brackets and the previous CLR.

Combinations inside brackets can be nested together up to the third level, and they can be opened with AND or OR.

**Example of contact logic representation (Ladder)**



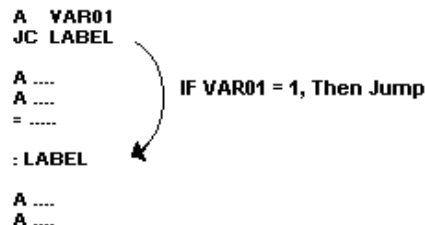
**Example of instruction list logic representation (AWL)**

```
A VAR01
A(
A VAR02
A(
O VAR03
O VAR04
)
O VAR05
)
= VAR06
```

## Jump to Labels

The Movicon logic programs are capable to manage instructions to jump to labels set up in the program. The jump instruction transfers the flow of execution to the preset labels (obligatory).

The jump instruction can be absolute or conditioned. In the first case the jump will be executed independently of the previous Combined Logic Result. In the second case, the jump will be executed only when the CLR = 1, otherwise the jump instruction will be ignored.



The jump label, which must always be preceded by the ":" character, can be either positioned at the front or at the back of the jump instruction.



**ATTENTION!** the programmer must take care not to insert closed jump loops otherwise this will block the General Logic with out any warning from Movicon.

## Variables in Floating Point

---

You can process logic by using the values in floating point. The variables or constants in floating point are 32 bit (Float).

To run programs in floating point, you need to precede the program with the **FLOAT** instruction. To restore work on integer values, you need to declare the **INT** instruction.

If not specified differently, the logic will assume work with integer values for default. It is therefore possible to run parts of the program working in floating point by enclosing the part of the program concerned between the "FLOAT" and "INT" instructions.

Example:

```

FLOAT

L KF 124,5
L TEMPER
*
T HT_TEMPERT

INT
```

## The Logic Accumulators

---

In the instruction in byte, word or doubleword, Movicon uses two system logic accumulators called ACCU1 to indicate the first accumulator (main) and ACCU2 to indicate the second accumulator. The accumulators are considered as temporary buffers used by the system to process and handle the variables in byte, word or doubleword. Only the ACCU1 is used in simple loading or transferring operations. ACCU1 as well as ACCU2 are used in comparison or mathematical operations.

Instruction	Value	ACCU1	ACCU2
L VAR00001	345	345	0
L VAR00002	89	89	345
+		434	89
T VAR00003	434	434	89



**Note:** the format of the data contained in the accumulators is understood as value with sign, independently of how the variable is declared in the DB. Therefore even the variables declared without sign are interpreted by the General Logic with sign.

**When working with data in Float, the accumulators work with variables in 32 bit floating point.**

## Constants

---

In the IL Logic you can use numeric constants preceded by the following suffixes:

KD	Integer decimal numbers (i.e. KD 28)
KH	Integer hexadecimal numbers (i.e. KH FE5A)
KF	Floating point numbers (i.e. KF 72.289)

Numerical constants may need to be used in compare instructions or in arithmetic instructions.

## End program processing

---

The Logic program ends automatically with the last instruction within it. The system returns back and executes the program from the beginning, cyclically, without needing a 'return' instruction.

Nevertheless you can insert an end program instruction which will terminate the processing and the return back to the beginning of the programming.

The end program instruction can also be conditioned by the CLR combined logic result.

```
A VAR00001
= VAR00002
BE          // end program processing unconditional

A VAR00003
BEC         // end program processing conditioned by VAR00003 variable
```

## Inserting Comments

---

We highly recommend that you use comments internal logic programs so that they or part of them are easier to understand. Movicon gives you unlimited use of comments in any point of the program, whether being General Logic or Local Logic.

In order to this you must first insert a double slash '/' at the beginning of the line, before typing the comment text. These characters indicate to the compiler to completely ignore the remaining text in the line.

Entering comments is made easier by using the Comment command from the Insert Menu from the 'Logic Explorer' window.

```
A VAR00001
S VAR00002 // ***** Enabling command Motor 1 *****
```



Comments can be inserted either at the beginning of the line, as headings for a block of instructions, or at the side of the instruction line. You must keep in mind that the text typed in at the point where '/' is inserted up to the end of the line will be considered as a comment.

# Logic Bit Instructions

---

The Logic in bit instructions allow binary logic variables, available in the system, to be combined in sequence with each other to create a CLR for activating following commands.

The start of a logic in bit sequence is done by directly using the AND or OR instructions.

## AND Logic

---

To combine two Movicon binary variables together in sets (boolean multiplication), you need to use the AND or AND NOT instructions. The AND NOT instructions will execute the AND operation with the associated variable's negation (contact closed).

Example: to set the "Out\_Motor1" output variable when both the "Start\_Motor" and "Cycle\_Machine" input invariables are at logic state '1', you need to:

```
A  Cycle_Machine
A  Start_Motor
=  Out_Motor1
```

## OR Logic

---

To combine in parallel two Movicon binary variables together (boolean sum) you need to use the OR or OR NOT instructions. The OR NOT instruction performs the OR operation with the associated variable in negative (contact closed).

Example: to get "Out\_Motore1" output when any one of "Start\_Motor" and "Cycle\_Machine" inputs are at logic status '1' logic, you need to do as follows:

```
O  Cycle_Machine
O  Start_Motor
=  Out_Motor1
```

## Sequence Combination

---

Logic sequences can be combined together. The example below illustrates the OR combination of two AND sequences for setting the output variable:

```
A  Ciclo_Auto
A  Start_Motor
O
A  Ciclo_Man
A  Start_Motor
S  Out_Motor1
```

## Setting Variables

---

The combination between two or more binary variables always determine a Combined Logic Result CLR. The combined logic result can determine the setting of one or more variables or command activations.

To set the binary state of a variable in function with the CLR, you need to use the '=' instruction. By doing this, the indicated variable will assume the value '1' when the CLR is equal to '1', or value '0' when the CLR is equal to '0'.

To assign a logic state to a variable in Latch mode, you need to use the SET (S) and RESET (R) instructions.

The Set instruction places a variable to logic state '1' when the CLR is equal to '1'. The variable will subsequently remain at logic state '1', even if the CLR changed to '0', until the Reset instruction of that variable is activated.

Example 1: the 'Out\_Motor1' variable will remain at '1' only when both the 'Cycle\_Machine' and 'Start\_Motor' variables remain at '1'

```
A  Cycle_Machine
A  Start_Motor
=  Out_Motor1
```

Example 2: the 'Out\_Motor1' variable goes to '1' only when both the 'Cycle\_Machine' and 'Start\_Motor' variables go to state '1' at the same time, after which it will remain at '1' until the 'Cycle\_Machine' variable turns to state '0', independently of the 'Start\_Motor' value.

```
A  Cycle_Machine
A  Start_Motor
S  Out_Motor1

AN Cycle_Machine
R  Out_Motor1
```

## Setting Timers

---

Movicon Timers start exclusively with the "=" instruction. The Movicon Timers are activation delay type. When the CLR is equal to '1', the set timer will start to count until the desired Preset time, in tenths of a second, has been reached.

When the timer is time count is higher or the same as that Preset, the Timer's variable ('Contacts') is set at the logic value '1' and will remain so until the Timer activation CLR remains at '1'.

```
A  Start_Timer
=  T  1.30      // timer preset with 3 seconds
```

There are 1000 timers available from 0-999. The Timers are 'local' logic, therefore each 'General' and 'Local' logic provide 1000 timers for use. This means, for example, that the '0 Timer' can be used internal the logic of each Movicon object without creating any malfunctioning.

## Summary Table Bit Instructions

Functions	Movicon Instruct.	Step7 ENG Instruct.	Step7 GER Instruct.	Operands	Descriptions
AND	A	A	U	I,O,F,T,Not Shared	Combination of AND logic between binary variables. Bit interrogation on signal "1"
OR	O	O	O	I,O,F,T,Not Shared	Combination of OR logic between binary variables. Bit interrogation on signal "1"
AND NOT	AN	AN	UN	I,O,F,T,Not Shared	Combination of AND NOT logic between binary variables. Bit interrogation on signal "0"
OR NOT	ON	ON	ON	I,O,F,T,Not Shared	Combination of logic OR NOT between binary variables. Interrogation of bit upon "0" signal."
LATCH	=	=	=	I,O,F,T,Not Shared	Setting of variable to the same value of the CLR. If the variable is a timer, it activates the time with activation delay.
SET	S	S	S	I,O,F,Not Shared	The variable is set to logic state "1" when the CLR = 1.
RESET	R	R	R	I,O,F,Not Shared	The variable is reset to logic state "0" when the CLR = 1.



Bracket AND	A(	A(	U(		Opens brackets. The contents of the brackets will be combined by AND with the previous CLR. The instruction always needs the close bracket ")".
Bracket OR	O(	O(	O(		Opens brackets. The contents of the brackets will be combined by OR with the previous CLR. The instruction always needs the close bracket ")".
JUMP Unconditioned	JU	JU	SPA	LABEL	Unconditional jump to specified label. The program flow will be transferred to the label position ":".
JUMP Conditioned	JC	JC	SPB	LABEL	Conditional jump to specified label. When the CLR = 1, the program flow will be transferred to the label position ":". When the CLR = 0 the instruction will be ignored.
Timer	= T 0.10				Start counting time (Delay) for a Timer.

# Word Logic Instructions

---

The word logic instructions allow the logic variables in byte, word or doubleword, which are available in the system, to be combined with each other in order to allow data up to 32 bits to be loaded, transferred or handled.



All the instructions in word can be conditionally executed according to the CLR, by placing the letter "C" after the instruction. In this way, when the CLR is equal to "1", the instruction will be executed or ignored if otherwise.



Instructions in word do not influence CLRs.

## The ACCU1 and ACCU2 Logic Accumulators

---

The instructions with byte, word or doubleword are provides the use of the Movicon Logic Accumulators. The main accumulator is called ACCU1 while the secondary one is called ACCU2. These accumulators are temporary buffers through which the system handles variables in byte, word or doubleword.

You must pay particular attention when using the instructions, described in the specific paragraphs, as to where the data is loaded, how it is processed and where the result is placed.

## Loading Variables and Constants

---

Before transferring variables or handling their contents, it is always necessary load them into the Accumulator. The load instruction or LOAD (L) just transfers the contents of the variable specified into the ACCU1 Accumulator. The system may have to move the contents of the ACCU1 to the second accumulator AACU2, if the ACCU1 already contains a value which must not be lost.

Apart from a variable the data to be loaded may also be formed by a numeric constant (preceded by a KD suffix), by a numeric constant with floating point (preceded by a KF suffix), by a hexadecimal (preceded by a KH suffix).

## Transferring Variables

---

When transferring previously loaded data or the result of previously run data processing, you need to use the TRANSFER (T) instruction.

This instruction evokes the transferring of the ACCU1 contents in the specified variable. The destination, cannot be a constant.

## Logic Operations on Word Variables

---

Movicon allows logic combinations of byte, word or doubleword variables. The AND logic or OR logic or OR EXCLUSIVE combinations permit the data of two variables to be combined (or of one variable and one constant) previously loaded in the accumulators, with the result placed in the ACCU1 accumulator. The result may they used as pleased before the accumulator is used again in logic.

Example: to execute the OR logic between two 'Data1', Data2' words and transfer the result to 'Data3' word, you need to do as follows:

```
// load Data1 value into ACCU1
```

**L Data1**  
**L Data2** // load Data2 value into ACCU1 and move the Data1 value into ACCU2  
**OW** // execute the OR between the two values and put it in ACCU1  
**T Data3** // transfers ACCU1 to Dato3

## Operations on Accumulators

Movicon allows the contents of two accumulators, ACCU1 and ACCU2 to be swapped over when needed by the programming requirements. When executing the **TAK** instruction, the contents in ACCU1 are transferred to the ACCU2 and viceversa.

You can also execute the binary complement to '1' of the data contained in ACCU1 by executing the **KEW** instruction. The result is always placed in ACCU1.

## Word Summary Table

Function	Movicon Instuct.	Step7 ENG Instuction	Step7 GER Instuction	Operand	Description
LOAD	L, LC	L, LC	L, LC	Byte, Word, DWord from any area. Constants KD,KH,KF	Loading a byte, word or doubleword into Accu1. Loading a numerical constant. LC instruction = loading conditional upon CLR
TRANSFER	T, TC	T, TC	T, TC	Byte, Word, DWord from any area.	Transfer of contents from Accu1 into a byte, word or doubleword. TC instruction = transfer conditional upon CLR
AND Word	AW, AWC	AW, AWC	UW, UWC		Data contained in two accumulators Accu1 and Accu2 combined by AND logic. The result is placed in Accu1. AWC = AND conditional upon CLR

OR Word	OW, OWC	OW, OWC	OW, OWC		Data contained in two accumulators Accu1 and Accu2 combined by OR logic. The result is placed in Accu1. OWC = AND conditional upon CLR
OR Exclusive	XOW, XOWC	XOW, XOWC	XOW, XOWC		Data contained in two accumulators Accu1 and Accu2 combined by logic EXCLUSIVE OR. The result is placed in Accu1. XOWC = EXCLUSIVE OR conditional upon CLR
ACCU Exchange	TAK, TAKC	TAK, TAKC	TAK, TAKC		Content of two accumulators Accu1 and Accu2 exchanged (Accu1 goes to Accu2 and vice versa). TAKC = Accumulators exchanged if CLR = 1
Complement 1	KEW, KEWC	INVI, INVIC	INVI, INVIC		Calculation of the Complement to "1" (NOT operation) on contents of accumulator Accu1. The result is placed in Accu1. KEWC = Complement to 1 upon CLR
BCD Convers.	BCD, BCDC	ITB, ITBC	ITB, ITBC		Data contained in

					Accu1 converted from binary code to BCD code. <b>The data must be in Word format.</b>
Decimal conversion	DCB, BCDC	BTI, BTIC	BTI, BTIC		Data contained in Accu1 converted from BCD code to binary code. <b>The data must be in Word format.</b>
Swap Byte	SWAPB, SWAPBC	CAW, CAWC	TAW, TAWC		Bytes exchange for the Accu1 Word
Swap Word	SWAPW, SWAPWC	CAD, CADC	TAD, TADC		Words exchange for the Accu1 DWord

# Compare Instructions

All the Movicon compare instructions always examine the ACCU1 and ACCU2 contents. The data contained in the ACCU1 or ACCU2 can be produced by processing operations or can be variables or constants previously loaded with the Load instruction.



The execution of a compare instruction will always determine a CLR Combined Logic Result.

Example: If you wish to transfer the 255 numeric constant to the "Data1" variable only when the "Data2" variable is equal to zero or on the contrary, if you wish to transfer the 0 numeric constant to the "Data1" variable you need to set as follows:

```
L Data1          // load Data1 value in ACCU1
L KD 0           // load zero in ACCU1 and move the Data1 value into ACCU2
==              // when Data1 = 0 , CLR = 1
L KD 255         // absolute loading of the 255 value into ACCU1
TC Data2        // when CLR = 1, transfer ACCU1 into Data2

L Data1          // load Data1 value into ACCU1
L KD 0           // load zero into ACCU1 and move Data1 value into ACCU2
><             // when Data1 is other than 0 , CLR = 1
TC Data2        // when CLR = 1, transfer ACCU1 into Data2
```

The compare operations are largely employed in all applications. The outcome of the comparison, which always determines a CLR, can activate any successive instruction or command type.

## Compare Summary Table

Function	Movicon Instruct.	Step7 ENG Instruction	Step7 GER Instruction	Operand	Description
Equal	==	==	==		"Equality" comparison between the data contained in the two accumulators. If Accu1 = Accu2, CLR is set at 1.
Different	><	><	><		"Difference" comparison between the data contained in the two accumulators. If Accu1 is different from Accu2, CLR is set at 1.
Greater	>	>	>		"Greater than" comparison

					between the two accumulators. If Accu2 is greater than Accu1, CLR is set at 1.
Lower	<	<	<		"Lower than" comparison between the two accumulators. If Accu2 is less than Accu1, CLR is set at 1.
Greater or Equal	>=	>=	>=		"Greater than or equal to" comparison between the data contained in the accumulators. If Accu2 is greater than or equal to Accu1, CLR is set at 1.
Lower or Equal	<=	<=	<=		"Lower than or equal to" comparison between data contained in the two accumulators. If Accu2 is less than or equal to Accu1, CLR is set at 1.

# Arithmetic Instructions

All the arithmetic instructions process and modify the contents of the ACCU1 and ACCU2 accumulators. The variables to be managed are therefore first loaded into the accumulators and, after the arithmetic instructions, transferred to destination variables. The execution of an arithmetic instruction does not influence the CLR Combined Logic Result.



The execution of an arithmetic instruction can be influenced by the CLR if the letter 'C' is placed after the instruction.

Example: Let's suppose that we need to transfer the result of an arithmetic subtraction between the "Data1" variable and the "Data2" variable to the "Data3" variable. To do this you need to set:

```
L Data1      // load Data1 value in ACCU1
L Data2      // load Data2 value in ACCU1 and move the Data1 value into ACCU2
-           // subtract the ACCU1 value from the ACCU2 value and put the result
           // into ACCU1
T Data3      // transfer ACCU1 to Data3
```

Arithmetic Instructions Summary Table

Function	Movicon Instruct.	Step7 ENG Instruction	Step7 GER Instruction	Operand	Description
Sum	+, +C	+, +C	+, +C		Arithmetic addition of the two values loaded in the accumulators (integers or floating point). Accu1 plus Accu2 with result in Accu1. +C instruction = Addition conditional upon CLR.
Subtraction	-, -C	-, -C	-, -C		Subtraction between the two numbers loaded in accumulators (integers or floating point numbers). Accu2 minus Accu1 with result in Accu1. -C instruction = Subtraction conditional upon CLR



Multiplication	*, *C	*, *C	*, *C		Multiplication between the two numbers loaded in accumulators (integers or floating point numbers). Accu2 multiplied for Accu1 with result in Accu1. *C instruction = Multiplication conditional upon CLR
Division	/, /C	/, /C	/, /C		Division between the two numbers loaded in accumulators. Accu2 divided by Accu1 with result in Accu1 (remainder in Accu2). \C instruction = Division conditional upon CLR.
Increment	I, IC			Byte, Word, DWord from any area.	Logic variable increment by specified quantity after point (e.g. to increment VAR0001 by 4, the syntax is = I VAR0001.4 ) IC=Increment conditional upon CLR.
Decrement	D, DC			Byte, Word, DWord from any area.	Logic variable decrement by specified quantity after point (e.g. to decrement VAR0001 by 1, the syntax is = D VAR0001.1 ) IC=Increment conditional upon CLR.
Increment ACCU1	INC	INC	INC		ACCU1 increment by specified quantity (e.g. to increment

					ACCU1 by 4, the syntax is = INC 4 )
Decrement ACCU1	DEC	DEC	DEC		ACCU1 decrement by specified quantity (e.g. to decrement ACCU1 by 4, the syntax is = DEC 4 )
Shift to Right	>>, >>C	SRD, SRDC	SRD, SRDC		Data bit contained in accumulator Accu1 are shifted to right by specified quantity. The freed bits are placed equal to zero (e.g. shift right by 6 places: >> 6 ). The right shift takes into account the sign of the value loaded in Accu1 (although variable used is without sign) setting the most significant bit at 1 if Accu1 is negative. >>C instruction = Shift to right conditional upon CLR
Shift to Left	<<, <<C	SLD, SLDC	SLD, SLDC		Data bit contained in accumulator Accu1 shifted left by specified quantity. The freed bits are placed equal to zero. (e.g. shift left by 2 places: << 2 ) <<C instruction = Shift to left conditional upon CLR.

# Mathematic Instructions

All the mathematic instructions process and modify the contents of the ACCU1 and ACCU2 accumulators. The variables to be managed are first loaded into the accumulators and then transferred into the destination after the mathematic instruction has been carried out.



If you wish to get precise mathematic results use values with floating commas by activating the "FLOAT" instruction before activating the mathematic instruction. The "FLOAT" or "INT" instructions cause the accumulators to swap over and remain active for the system until an instruction to the contrary is given.

All the instructions can be made conditional upon the previous CLR. By placing the letter "C" at the end of the instruction it will be executed only when the CLR is equal to "1", otherwise it will be ignored.

The execution of a mathematic instruction does not influence the CLR.

Example: if you want to transfer the calculated cosine of the number contained in the "Data1" variable to the "Data2" variable you will have to set as follows:

```
Float      // preset to work with floating comma
L Data1    // load Data1 value into ACCU1
cos        // execute cosine calculation and put it into ACCU1
T          // transfer ACCU1 to Data3
Data3
int        // preset to work with integers
```

## Mathematic Instructions Summary Table

Function	Movicon Instruct.	Step7 ENG Instruction	Step7 GER Instruction	Operand	Description
Floating Point	FLOAT	DTR	DTR		Preset for calculations with numbers in floating point format. The following mathematical numbers will be considered in floating point format. This is a modal instruction and can be deactivated using INT instruction.
Integer	INT	RND	RND		Preset for calculation with integers. The following mathematical numbers will be considered

					integers. Active by default unless otherwise specified.
Sin	SIN, SINC	SIN, SINC	SIN, SINC		Calculation of Sine of number contained in Accu1. The result is placed in Accu1. The value of the function is express in Radiant.
Cosin	COS, COSC	COS, COSC	COS, COSC		Calculation of Cosine of number contained in Accu1. The result is placed in Accu1. The value of the function is express in Radiant.
ArcSin	ASIN, ASINC	ASIN, ASINC	ASIN, ASINC		Calculation of ArcSine of number contained in Accu1. The result is placed in Accu1. The value of the function is express in Radiant.
ArcCosin	ACOS, ACOSC	ACOS, ACOSC	ACOS, ACOSC		Calculation of ArcCosine of number contained in Accu1. The result is placed in Accu1. The value of the function is express in Radiant.
Tangent	TAN, TANC	TAN, TANC	TAN, TANC		Calculation of Tangent of number contained in Accu1. The result is placed in Accu1.

					The value of the function is express in Radiant.
ArcTangent	ATAN, ATANC	ATAN, ATANC	ATAN, ATANC		Calculation of ArcTangent of number contained in Accu1. The result is placed in Accu1. The value of the function is express in Radiant.
Hyperbolic Sin	SINH, SINHC				Calculation of Hyperbolic Sine of number contained in Accu1. The result is placed in Accu1. The value of the function is express in Radiant.
Hyperbolic Cosin	COSH, COSHC				Calculation of Hyperbolic Cosine of number contained in Accu1. The result is placed in Accu1. The value of the function is express in Radiant.
Hyperbolic Tangent	TANH, TANHC				Calculation of Hyperbolic Tangent of number contained in Accu1. The result is placed in Accu1. The value of the function is express in Radiant.
Exponential	EXP, EXPC	EXP, EXPC	EXP, EXPC		Exponential calculation of natural number "e" with exponent contained in

					Accu1. The result is placed in Accu1.
Power	POW, POWC				Exponential calculation of Accu1 raised to the power of Accu2. The result is placed in Accu1.
Logarithm "e"	LOGE, LOGEC	LN, LNC	LN, LNC		Calculation of natural logarithm (base "e") of number contained in Accu1. The result is placed in Accu1.
Logarithm 10	LOG, LOGC				Calculation of base 10 Logarithm of number contained in Accu1. The result is placed in Accu1.
Square Root	SQRT, SQRTC	SQRT, SQRTC	SQRT, SQRTC		Calculation of square root of number contained in Accu1. The result is placed in Accu1.

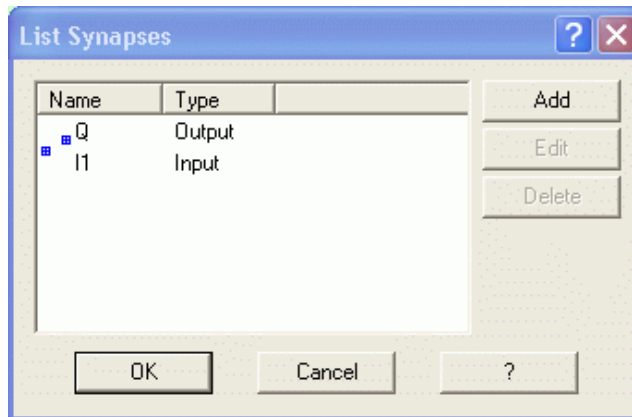


Each object can be configured with a determined number of inputs and outputs (the synapse), which will be linked to other symbols using the connector objects. The input and output status can be read within the basic script code of each object and the appropriate output values can be written as a consequence and propagated to the other graphically linked objects.



## Inserting Synapse

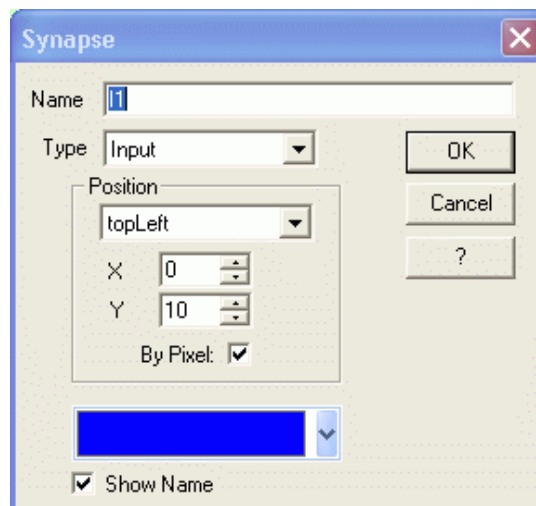
The first thing to do when creating Synapse logic is to assign the screen object one or more synapse, in "Input" or "Output". To insert an object's synapse right click on the object and select the "**Edita Sinapses...**" command to open the dialog window listing any already Synapse associated to the object:



The following commands can be used through this window:

- **Add**: this button adds a new synapse, by associating it to the selected object. Upon pressing this button another dialog window will open to allow you to configure the synapse as described below.
- **Edit**: this button allows you to edit the synapse selected in the list. Upon pressing this window the same dialog window used for inserting a new synapse will show as the editable properties are the same ones at this moment.
- **Delete**: this button allows you to delete the synapse selected in the list and therefore its association to the object selected.

As mentioned above, the "Add" button is used for inserting a new synapse which will be configured to define its type and graphical aspect. Once a synapse has been created it can then be edited afterwards using the "Edit" command. The Synapse configuration window is as shown below:



**Name**

This editbox is used for assigning the synapse with a name. Its name must be unique and different from the other synapses defined for the object you have selected. An error message will appear, If you insert a name that has been already given to another synapse.

**Type**

This selection box permits you to establish whether the synapse should be input or output:

- **Input:** the synapse is considered as an input point for the object in which it is inserted and can be connected to other output synapse in the same screen.
- **Output:** the synapse is considered as an output point for the object in which it is inserted and can be connect to other input synapse in the screen.

**Position**

These controls are needed for establishing the synapse's position, presented by a small colored dot, within the object in which it has been inserted. The position is provided with a selection dot in which to apply the synapse and any eventual offset in the X and Y coordinates for moving the dot within the object. The position options are:

- **top left**
- **top right**
- **bottom right**
- **bottom left**
- **at the top**
- **on the right**
- **at the bottom**
- **to the left**
- **in the middle**

You must consider that the synapse's initial position is always on the edge of the rectangle that surrounds the object it is associated to. However, you can associate a movement offset starting from this initial position:

- **X:** this editbox establishes the synapse's horizontal movement. This movement is always towards the right with positive numbers and towards the left with negative numbers, independently from the position assigned to the synapse.
- **Y:** this editbox establishes the synapse's vertical movement. This movement is always towards the bottom with positive numbers and towards the Top with negative numbers, independently from the position assigned to the synapse.
- **by Pixel:** The synapse's offset will be calculated in pixels when this check box is enabled.



It is not possible to position the synapses outside the symbol's perimeter (of the surrounding rectangle). In cases where a X and Y offset has been assigned making the synapse go outside the object, it will still be kept inside.

**Back Color**

This button is used for assigning the synapse a color. When clicking this button a color will show enabling you to pick the one desired. By assigning different colors to the synapses allows you to visually tell Input and Output synapses apart.

**Show Name**

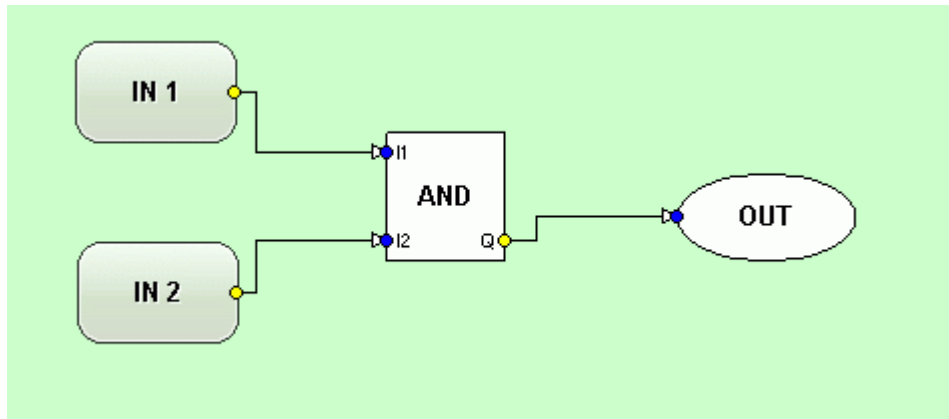
This option box is used for displaying the name of the assigned synapse at the side of the small coloured circle. The name's position follows the position assigned to the synapse and the text is positioned inside the symbol in which the synapse has been associated.

## Connecting Synapses

---

Synapses can be connected to each other within screens by using "Connector" objects selectable from the Toolbox "Advanced Shapes" group. These objects are displayed within the screen as connector lines and their properties follow the synapses they are connected

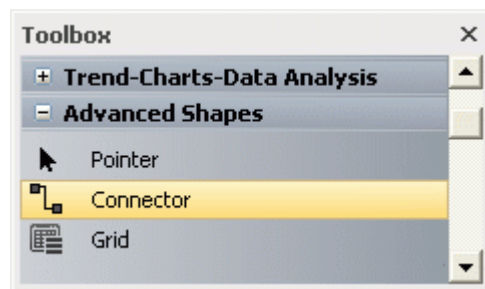
to. Therefore once the synapse are connected up together, they can be moved around within the screen without losing their original connections.  
This diagram below shows connected synapse objects. The "Out" synapse are in yellow and the "IN" synapses are in Blue.



If the objects are moved, their Connector lines will change position and shape accordingly.  
Below is described how the synapse connectors are inserted or modified:

#### Adding Connectors

The "Connector" objects are available from the Toolbox "Advanced Shapes" group. Unlike other objects, these objects remain valid while being inserted until another object type is selected or restoring the "Pointer" in the toolbox. You will find that this will speed up synapse connecting operations of the various drawings and symbols on screen.



A new connector must always start from an Out synapse and finish at an In synapse. A grey square will appear within the synapse to indicate that it is selectable, on the contrary a no entry sign will appear to indicate that the Connector cannot be connected at that point, or that there is no synapse, or the synapse is not the input/output type consented at that moment.

It is important to remember that each synapse can be connected with more than one connector and therefore more than one object. This allows one single out synapse value to be shared with more objects at the same time, simplifying code which controls logic.

#### Modifying a Connector

Modifying a connector inserted on screen can be done by simply selecting it and dragging one of its ends to another synapse of the same type (input/output). In order to do this, you need to first restore the point if not already done so, using the "Pointer" command from the toolbox. After having clicked on the connector line to be changed, if you move on to one of the two synapse connected, the mouse pointer will change shape indicating that the connector can be dragged. Press and keep pressed the mouse key to drag the connector end to another synapse.

#### Deleting a connection

In order to delete a connection restore the mouse pointer, if not done so already, using the 'Pointer' command from the toolbox. Select the connector to be deleted and cancel it with the "Canc" keyboard key or use the "Delete" command from the Edit menu which appears with a right mouse click.

Deleting an object containing connected synapse will automatically delete all of the Connectors in the object.

### Connector Properties

A Connector object is a normal line with all the effects and properties available for this type of object. Therefore, please refer to the relating section describing object properties. To modify the connector properties, select the one desired and then use its property window.

## Synapse Logic Programming

So far we have seen how to insert and link synapse with each other within a screen. Now we come to the most complex stage: programming Synapses.

The relationship between an input synapse and an output synapse must be programmed within the basic script code of the object to which the synapses are associated and for which you will need basic script programming know how.

Below you will find a brief description on the properties, methods and events regarding the synapse basic script programming. For further information on these functions and basic script programming we suggest you refer to the sections on VBA languages.

In cases where the user is not an expert basic script programmer but wishes to use synapses in a screen, they can use symbols from the Movicon Symbol Library and build a logic linking the various synapses.

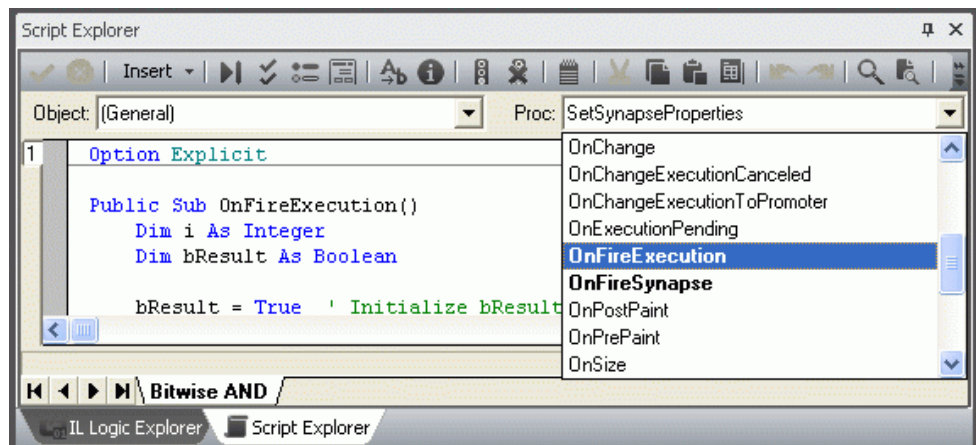
To access the basic script code of each object, select the object and activate the "Script Explorer" window.

### Events generated from synapse executions

In the basic script code of each symbol you can use a series of events, generated automatically by the system the moment in which the synapses are executed within a screen.

Each event is generated in a determined moment, during the synapse logic execution and is listed below.

A new event is entered by selecting the corresponding item in the "Proc" dropdown on the top right of the "Script Explorer" window. In cases where the event has already been managed, the script will appear in bold on the list will appear in bold and its selection will place cursor focus on the same tested procedure.



The events available for managing synapses are:

OnChangeExecutionCanceled (DrawCmdTarget Event)  
OnChangeExecutionToPromoter (DrawCmdTarget Event)  
OnExecutionPending (DrawCmdTarget Event)  
OnFireExecution (DrawCmdTarget Event)  
OnFireSynapse (DrawCmdTarget Event)

OnStartSynapsisExecution (SynopticCmdTarget Event)  
OnStopSynapsisExecution (SynopticCmdTarget Event)

## Synapse Methods and Properties

---

There are a series of methods and properties available which allow you to create basic script code for executing synapse logic. As already explained in the section on using VBA code, these methods and properties are part of the "DrawCmdTarget Property" and "SynopticCmdTarget" basic script interface.

### Properties:

SynapseBackColor (DrawCmdTarget Property)  
SynapseValue (DrawCmdTarget Property)  
SynapseValueFromID (DrawCmdTarget Property)  
SynapsisVisible (DrawCmdTarget Property)

SynapsisExecution (SynopticCmdTarget Property)

### Methods:

GetConnectorObjectConnected (DrawCmdTarget Function)  
GetNumConnectionsOnSynapse (DrawCmdTarget Function)  
GetNumSynapsis (DrawCmdTarget Function)  
GetObjectConnectedOnSynapse (DrawCmdTarget Function)  
GetSynapseName (DrawCmdTarget Function)  
GetSynapsePoint (DrawCmdTarget Function)  
HasSynapsis (DrawCmdTarget Function)  
IsSynapseConnected (DrawCmdTarget Function)  
SynapsePassExecution (DrawCmdTarget Function)

GetObjectByUniqueID (SynopticCmdTarget Function)  
SetSynapsisVisible (SynopticCmdTarget Function)

## Synapse Execution

---

Synapse programmed within a screen can be executed at any given time using the appropriate commands available from the different Movicon resources.

Synapses are executed starting with the first object's **OnFireExecution()** event contained by the synapse logic inserted in the screen. The same events are generated for the other objects based on the order they have been connected using the Connector objects, and commanded by the "**SynapseValue()**", "**SynapseValueFromID()**" properties or the "**SynapsePassExecution()**" method.

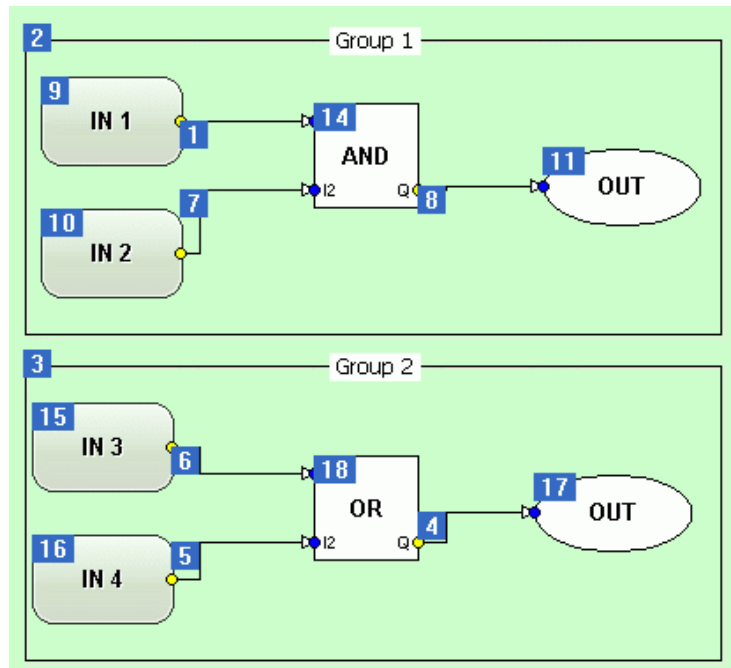
When these above mentioned properties or methods are not used in an object's **OnFireExecution()** event, the synapse logic's execution will remain pending and still in that object.

More than one distinct synapse logic block can be inserted on screen, where each block will be composed of objects and synapse connected to each other with connectors. However, the individual block cannot be connected to other but you can establish a tab order to execute each block when needed.

The order in which the synapse logic is executed is established through each object's tab order settings, activated with the "**Tab Order**" from the **Layout** menu or with the "**Ctrl+D**" keys.

In the flow chart illustrated below a screen is displayed with two synapse groups, marked out in two decisive group boxes.

When the synapse logic execution command is given, the logic group in "Group 1" with the AND block will be executed first. This is because the connector object with the lowest tab number is in Group 1.



It is clear from this chart that only one connector object belonging to one group has been set in a tab order to allow quicker order selecting. The final execution flow is therefore determined by the logic connections created between the various synapse, but is given execution priority over those groups containing connector objects assigned given lower tab orders compared to the others.

The command to start the synapse logic execution can be given in different ways as follows:

#### Synapse Logic Execution from Screen

There are two execution properties available in the screens, which consent you to manage the synapse logic execution start:

- **Execute Synapse:** When enabled, this property will activate the execution of any synapse logic upon Screen load. The logic will only be executed once.
- **Synapses Cyclic Execution:** When enabled, this property will activate the cyclic execution of any synapse logic when the Screen is active. The synapse logic needs to be executed at least once before the cyclic execution activates, therefore you will need to enable the "Execute Synapse" property as well or invoke the start command through one of the other methods described below.

#### Execute Synapse Logic on command

The synapse logic execution can be commanded using the Movicon Command List. In the "Screen" command group you will find the "Execute Synapses" command which will activate the synapse logic execution for the screen selected. The Command List can be used in different Movicon resources and therefore synapse logic execution activation can be commanded as pleases. In addition, it is absolutely necessary that the screen be open or loaded in memory in order for its synapse logic to be executed.

#### Execute Synapse Logic using Basic Scripts

To start the synapse logic execution from a basic script you will need to use the appropriated "ExecuteCommand" function. The command string to be passed to the function is:

```
ExecuteCommand("<CommandType synoptic='Screen1' action='6' monitor='0'
parameter="" x='-1' y='-1' width='0' height='0' Caption='true' Border='true'
Resizable='false' SysMenu='false' MinimizeBox='false'
MaximizeBox='false'>Synoptic</CommandType>")
```

It is important to specify the below parameters for this type of command:

synoptic: name of the screen where the synapse logic is to be activated  
action: The value 6 identifies the "Execute Synapse" command

It is indispensable that the screen be opened or loaded in memory in order to execute the synapse logic.

### **Synapse Visibility**

---

You may find it useful not to have the synapse and their connectors visible on screen during the runtime mode. This can be done by disabling the screen's "Show Synapses" style property, or by using the appropriated "SetSynapsisVisible" basic script function.

## 26. Soft Logic IEC 1131

---

***Movicon integrates an optional programming environment called Logicon, a SoftPLC software compliant to the IEC 61131-3 standard.***

Logicon is a SoftPLC programming environment integrated in Movicon. Thanks to this powerful tool, users are now fully equipped with an all-in-one environment covering supervision and RealTime I/O logic control.

Logicon is the result of intense collaboration between Progea and KW Software GmbH, the German MultiProg producer company, whose Logicon version is integrated in Movicon.

The advantages offered are numerous where only one unique platform is needed, capable of developing and creating both on the HMI side and control side, with the enormous benefit of sharing Tags in an all-in-one system. Projects are run synchronized but separately in two Windows' layers (User Mode for supervision and Kernel Mode for PLC).

The Movicon development environment is now even more enhanced and enriched with the additional PLC IEC 61131-3 planning environment. The PLC runtime, called ProConOs, is executed in a realtime kernel as a normal PLC and is also available for both Windows XP/Vista and Windows CE.



The Soft Logic programming environment on-line help is only available in English and can be accessed through the Soft Logic editing window.

### Soft Logic Software Components

---

In order to create a project with Soft Logic you will need to use certain components to execute it in runtime and exchange data with Movicon. These components are:

- **Logicon:** is a soft logic project development which has been implemented in Movicon and therefore can be opened as a Movicon MDI window.
- **ProConOs:** is a soft logic runtime module. Once the project has been created it must be compiled and transferred to this module for running. ProConOs is available for both Win32 and WinCE versions.
- **ProConOs OPC Server:** is the soft logic's OPC Server which consents tags to be exchanged between the ProConOs runtime and Movicon.

### Integrated Soft Logic Editor

---

The SoftLogic Logicon editor has been integrated in Movicon by means of using a resource with the same name "Soft Logic". This resource appears in the Project Explorer's tree structure and has specific commands and configuration options displayed in its corresponding properties window. It also comes with an editor which is opened by double-clicking the resource name.



**In cases where other Movicon processes have been started up (Movicon.exe, MoviconService.exe or MoviconRuntime.exe) you won't see the "SoftLogic" resource listed in the project's Explorer window.**

As already mentioned, by double-clicking on the Soft Logic resource will open its editor. If this editor has not yet been opened for creating a soft logic project, a window will show for selecting the Template to be used for beginning the project. If you already have a soft logic project it will open up directly in editing mode.



The soft logic project will be created with the same of the Movicon project but with the ".mwt" extension. In addition, a new folder will be created with the same name of the project within which all the soft logic project files will be inserted.



**The ".mwt" project file and the soft logic folder must have the exact same name of the Movicon project.**

There are two "Soft Logic" resource command selections available in the "Command" pane in the "Project Explorer" window:

#### **Soft Logic OPC Server Configurator**

This command allows you to open the ProConOS OPC Server configuration window. The ProConOS OPC Server is the channel which consents Movicon application tag communication with Soft Logic. The OPC Server must therefore be configured to allow Movicon projects and the ProConOS SoftLogic to communicate with each other correctly (see "ProConOs OPC Server Configuration").

#### **Synchronization Variable Database**

This command allows the Soft Logic tags to be synchronized with those of the Movicon project. Tags can be synchronized each time modifications are done to the global tag table in the Soft Logic. This synchronization consents missing tags to be imported within the Movicon project and, for each tag imported, the dynamic link is set automatically to the tag in the Soft Logic (see "Variable Database Synchronization").

### **Automatic Soft Logic Commands**

---

Movicon manages the following Soft Logic commands in automatic while working in project development mode:

- **Soft Logic Compilation:** this command is activated each time the Movicon project is saved only on the condition that the Soft Logic editor be previously opened at least once in the opened project.
- **Soft Logic Compilation, Stop, Upload and Restart:** this command is activated every time the Movicon project switches from design to run mode only on the condition that Soft Logic editor be previously opened at least once in the opened project. Movicon manages a 10 sec. timeout when uploading the SoftLogic after which the project is started even when no Upload termination notifications have been received from the Soft Logic.

### **Debug Window in Runtime mode**

---

When the Movicon project is activated in runtime from the development mode, you will find a button in the command runtime bar to open the Soft Logic Debug window. This window is used for carrying out modifications to the Soft Logic project, while the Movicon project is being run, which are applied by uploading them to the Soft Logic PLC ProConOS. In addition, the soft Logic editor button also allows you to activate the debug mode, consenting you to read the status of the tags in the Soft PLC ProConOS, force values and monitor POU logic executions.

### **Soft Logic General Properties**

---

These general properties are used for setting parameters for configuring the Soft Logic OPC Server.

To edit this properties, select the Soft Logic resource and use the Movicon **"Properties Window"**.

#### **OPC Server Soft Logic**

This edit box allows you to insert the ProConOs OPC Server name to be used for connecting to the Soft Logic (default = "OPC.OUT.PCOS.21").



Modifications to the OPC Server name do not require you to re-enter or modify the dynamic addresses of each already imported tag. Modifications will be displayed automatically and applied when the project is saved.

#### OPC Server Resource Name SoftLogic

This edit box permits you to insert the name of the OPC resource to be used for connecting to the Soft Logic. The ProConOS OPC Server Configuration window consent you to insert more than one resource. Each resource can be set connect to a different Soft Logic or PLC. The name or the resource must be the same one set in the ProConOS OPC Server.



Modifications to the OPC Server name do not require you to re-enter or modify the dynamic addresses of each already imported tag. Modifications will be displayed automatically and applied when the project is saved.

#### Default Variable Group Name

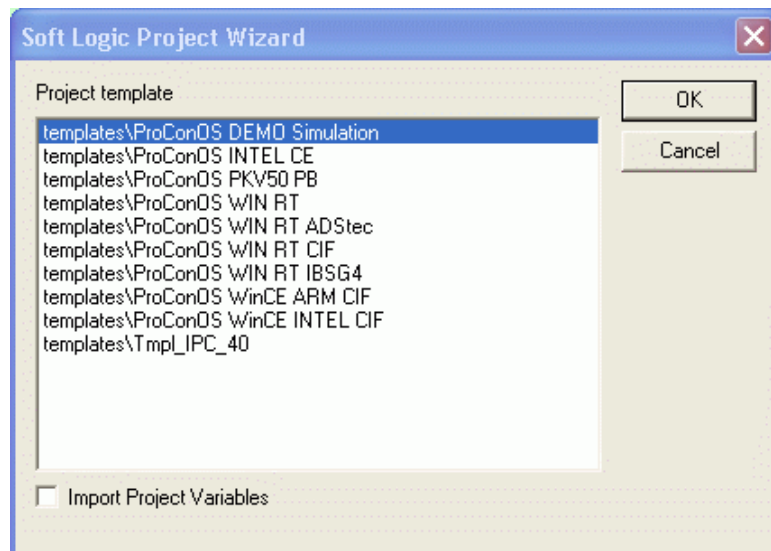
This edit box permits you to customize the default name to be used for the Movicon Tag group which contain the Soft Logic tags. Other sub-groups will be created within this group reflecting the same group hierarchy in the Soft Logic.

## Variable Database Synchronization

The Soft Logic tags can be synchronized with those from the Movicon project. This synchronization is bidirectional, but there are few restrictions that must be taken into account. Below is described the procedures to use for synchronizing tags.

#### Importing Movicon RealTime DB tags to Soft Logic

Tags created in the Movicon project, kept in the RealTimeDB, can be imported to the Soft Logic project only when the project is created first time round where a window will show for selecting the template to be used for creating the Soft Logic project. In this window you will find the "Import Project Variables" option. When enabled this option consents the Movicon RealTimeDB variables to be imported to the Soft Logic project when enabled.



When importing tags automatically, Movicon will insert the OPC link to the Soft Logic tags in the in the "Fixed I/O address" tag properties. During this phase you will not be able make tag selections to be imported as all tags existing in the RealTimeDB will be imported. Once the Soft Logic project has been created you will not be able to import other Movicon variables.

#### Importing Soft Logic tags to the Movicon RealTimeDB

Importing soft Logic tags to Movicon can be done at any time using the "Synchronize Variable Database" command from the "Project Explorer Window's command pane. This command gets enabled automatically each time a tag's "OPC" option is enabled in the Soft

Logic "Global Variables" resource. If this command should become disabled, just enable the property of any Soft Logic variable or check this option again. "Synchronize Tag Database" command activation will import all those Soft Logic tags with enabled OPC and not already present in the Movicon RealTimeDB. The imported tags will be created with the same names of those from the Soft Logic and with the "Fixed I/O address" properties already compiled with the corresponding OPC links. These tags will then be inserted in a tag group with the same name set in the "Nome Gruppo Tag di Default" property. Other sub-groups will be created within this group reflecting the same group hierarchy in the Soft Logic.

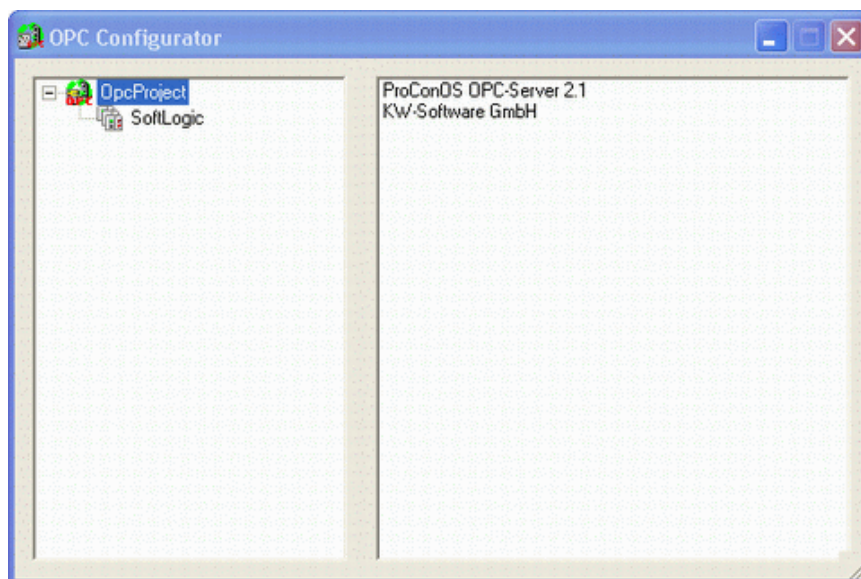


**New tags can be created in the Soft Logic using the "Data Types" resource. These types of tags can be "ARRAY", "STRING" and "STRUCT". The tags which have been declared with a "Data Type" in a custom "Global Variables" resource will not be imported automatically to the Movicon RealTimeDB.**

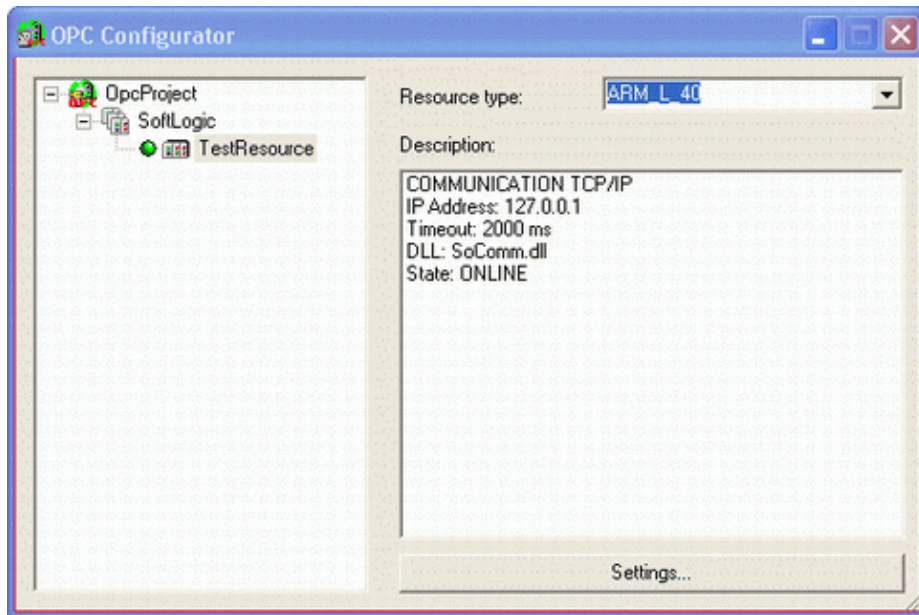
## ProConOs OPC Server Configuration

Once the Soft Logic project has been created you should proceed in configuring the OPC ProConOs Server to associate the Soft Logic as follows:

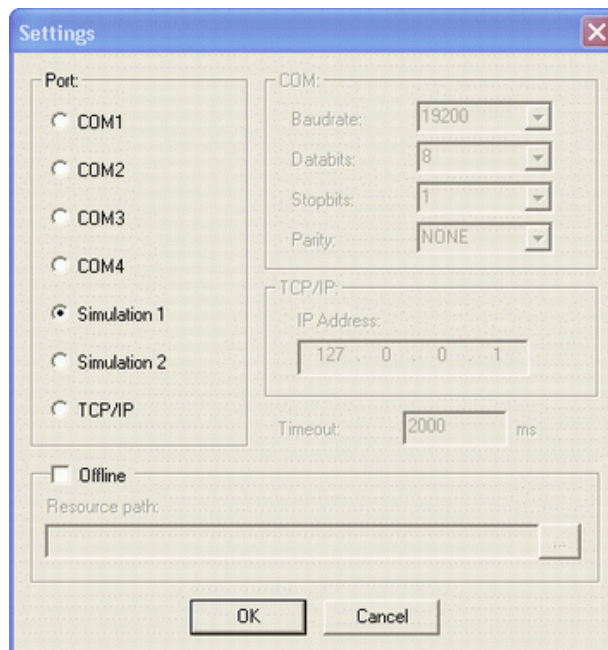
1. Start the OPC configurator using the "Soft Logic OPC Server Configurator" command found in the "Commands" drop down window at the bottom right of the "Project Explorer" window, or from the Windows "Start - All programs - KW Software - ProConOS OPS server 2.1 desktop - OPC Configurator" menu. This configurator window will appear:



2. Right click on the "OPCProject" and select "New Workspace..." or "Open Workspace..." if you have a workspace already. This will open a browse window where you must select the Soft Logic project in question. This will change the name of the "Workspace" to the project's name.
3. Now right click on the project's name (Workspace) and assign a new resource using the "New Resource..." command. Change the name of the new resource by inserting the one defined in the Movicon "OPC Server Resource Name SoftLogic" property (or as an alternative you can insert a name of your choice and then also insert it in the Movicon "OPC Server Resource Name SoftLogic" property. Keep the resource name selected and select the "ARM\_L40" item from the "Resource Type" list on the right.



4. Click on "Settings..." button to open the dialog window below and select the communication port option desired. If you don't have any field devices you can select "Simulation1" to test the project in simulation mode. Otherwise select "TCP/IP" if communicating through Ethernet and specify the device's IP address. Confirm with OK to close the window to go back to the above window.



5. Right click on the "OpcProject" item and invoke the "Save Workspace" command. This will create a file with the resource name in the Soft Logic project folder (ie. "TestResource") with the ".opc" extension containing the OPC settings. This file will be the one used by the ProConOs OPC Server for exchanging data with the field devices and the Movicon OPC Client.
6. In cases where a Windows CE device has to be used and therefore the OPC Server, the ProConOs will be executed on the WinCE device, the ".opc" file ("TestResource.opc" as an example) must be copied to the "OPCProjects" folder from the OPC Server ProConOs installation folder in the WinCE device.



# 27. Child Projects

---

*Movicon allows you to structure projects by decentralizing resources in other projects (child) with dynamic relationships, giving you the possibility to distribute your projects.*

Movicon has a powerful and innovative feature, which allows you to face new challenges in planning supervision systems. "**Child Projects**" are normal Movicon projects which, even when planned to function independently, are linked to parent projects creating a "Parent-Child" relationship where the Parent project is provided with all the resources of the Child project as if they were its own.



One project can be associated with many Child projects which can then become Parent projects and have their own Child projects. Therefore you can create a true and real dropdown family tree of projects.

This opens the way to many types of advantages. Let's look at the main ones:

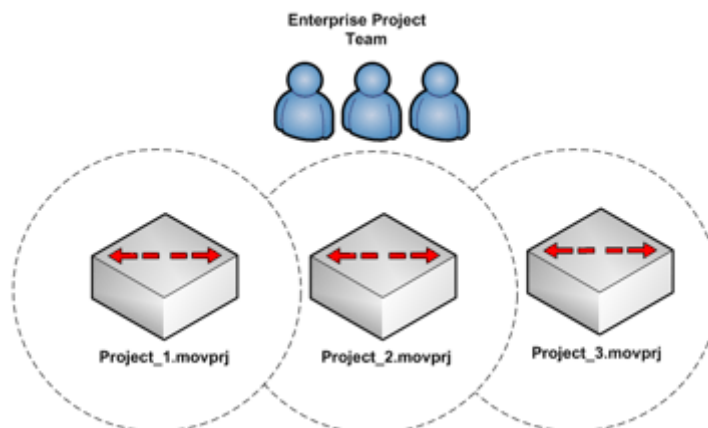
## Distributed Projects

---

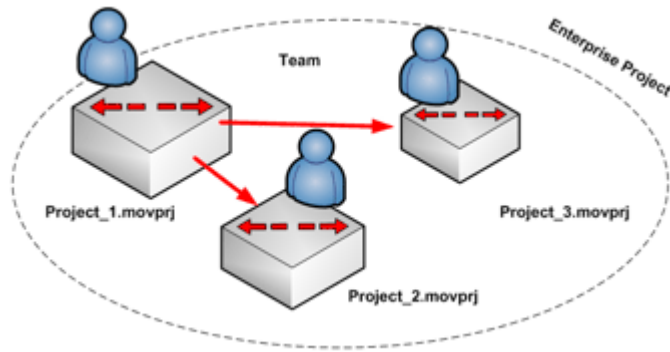
Projects structured with Parent-Child relationships provide many advantages to companies working in teams. In respect to the traditional technologies, where many people work using and sharing tasks in the same project, Movicon offers the possibility to distribute work in different projects independently where the Team Leader can have, in their own Parent Project, all the resources of the Child Projects of its collaborators, who can also completely work independently.

The Father Project is provided with all its children's resources internally, without any resource name distinction or duplication, as the name difference is governed by the child project's path. Therefore, for instance, a VAR0001 may co-exist in the parent project as well as in the child project because individualized by the project's name and path.

- Note: This architecture provides the automatic startup of child projects in Run when the Parent project Run is started up. This option is found in the Project's "Child Project Options Settings" properties.



A traditional Project team structure diagram



A distributed project structure diagram of Parent-Child project relationships

## Distributed Run

The Father-Child Project relationship is very useful for modular systems or machines where, for instance, the plant is divided into zones which can also be independent from one to another. In a situation like this you can create more projects, one for each zone, and then integrate them into one Father project from which you can access the pages and the variables of the Child projects.

### Example:

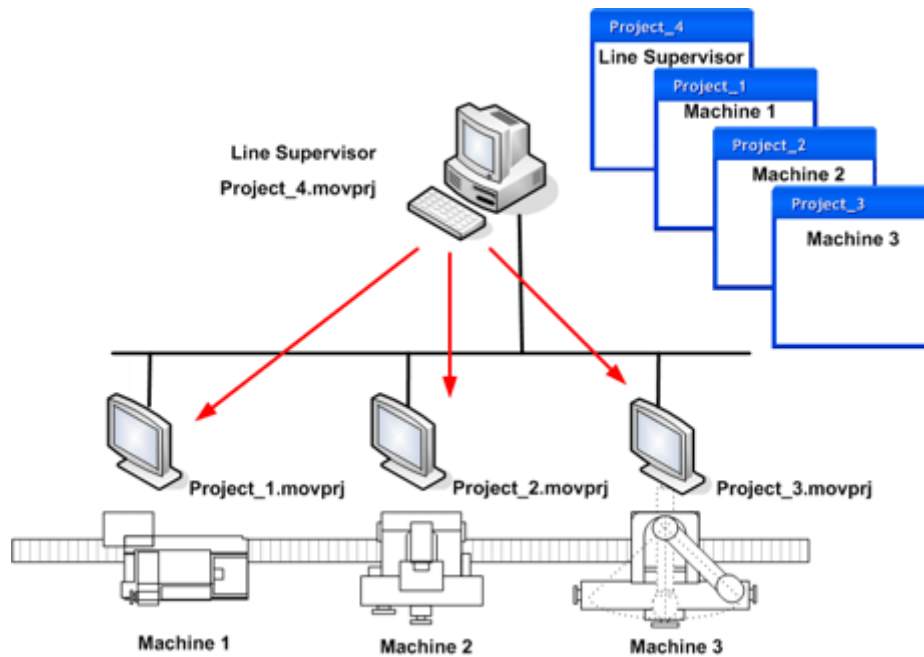
An automation line is composed of 3 independent machines. Each machine has its own project run locally on its PC. The machines are then integrated into a production line and linked to a main Supervision PC.

The big advantages Movicon gives you not only involve drastic development time reductions but also the chance to create **a main supervision project such as the "Parent Project" and three "Child Projects" representing the three individual machines**, which reside in local PCs.

In this way, the parent project can automatically be provided with all the individual variables of the various child projects, to produce general summary screen layouts. By using these general layout screens, residing in the parent project, **you can then open the screens of each individual machine** by simply opening the child project screens, which reside locally in the machines' PCs, in the father project. Not only do you save time but you get the advantage of having any **future modifications executed on the machines will automatically be executed in the main supervisor as well**.

- Note: In the example indicated for this architecture, the automatic child project startup with the Parent project run startup, option provided in the Project "Child Project Options Settings" properties, is not required (being independent projects). However, the child projects must be entered with the Server project's (Supervisor) IP address in their settings.





This figure illustrates an automation system composed of one Server project (Supervisor) being the "Parent Project" of three individual local machine projects being the "Child Projects"

### Client-Server with Child Projects

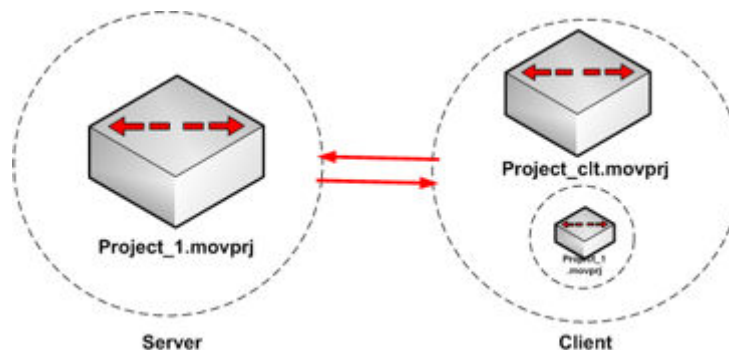
Parent-Child Projects may also be very useful in situations when needing a Server station and a certain number <n> of the "same" Client projects.

In this case, by using the "Parent-Child" relationship you can create child projects which can be easily be distributed in any PC network that won't need modifying if the Server has to be modified.

In order to do this just create an 'empty' Client project and setup a " project within to correspond to the Server project.

The child project can reside in the Client physically (by making an exact copy of the Server project beforehand) or it can be directly linked to the Server project via net. **This second configuration will allow you to use the same identical Client project in any PC connected in net.** Any client can connect to the Server PC to run project files.

Furthermore, this technique automatically provides you with software centralization: **any modifications executed on the Server will automatically be available to all clients.**



Client-Server structure with Parent (Client) - Child (Server) relationship



- Note: In the example indicated for this architecture, the automatic child project startup with the Parent project run startup, option provided in the Project "Child Project Options Settings" properties, is not required (being independent projects). **However, child projects must be entered with the Server project's (Supervisor) IP address in their settings.**
- **In this context it is important to specify that the port number set in the parent project's networking service is different from the one set for the child projects.** For example, the child projects can all use the port 12233, but the Parent project must be configured on a different port, for example 12234. If you don't do this, you may have some problems when a Web Client or a Client needs to be connected to the Parent project. In fact, if you open a child screen from the Parent project, and since the screen's variable has to be connected to the network server configured within the child project, Movicon runs the network services for that specific child project. If the child project uses the same port as the parent, conflicts will certainly occur.

## Child Project Paths

---

The 'Child Projects' can be retrieved both with a local PC path in which the 'Parent Project' resides and with a network path.

When the Child project is local, it would be to your advantage to insert the folder containing the "Child Project" in the "Parent Project" folder, even though in reality it should be possible to set any path. In this way, however, it makes it easier because the search path of the "Child Project" will always be related to that of the "Parent Project", therefore by moving all of the "Parent Project" folder in a different path or to another PC, will not cause any problems of unfound absolute paths.

When the "Child Project" is in the "My Documents" folder the search is always carried in relation to the "My Documents" folder of the user logged on to the operating system at that moment.

When the 'Child Project' is in another folder or in a computer working in net, the search path is a fixed one. In this case it is best to emphasize that the folders from the PC are to be shared.



When possible it is always best to insert the 'Child Project' inside the 'Parent Project' folder or subfolder to avoid absolute search paths errors.

## Child Project Resource Access

---

One of the great uses of using 'Child Projects' is that they have the same resources of their 'Parent Project' which can be accessed from both projects. For instance a 'Parent Project' may need to call up a screen of the Child Project and vice-versa.

The pages of both projects may be displayed during Runtime without realizing that they come from different projects.

In the programming phase, resource access between Parent and Child projects simply requires only one syntax. Normally in the browse window for selecting the Parent project's resources you can also see the Child project's resources. This makes it easier to select the resource desired. However the Parent project's resources cannot be seen in the Child project therefore it is necessary to type in the name of the interested resource.

The syntax needed for accessing the Child Project's resources from the Parent project is as follows:

**ChildProjectName\ResourceName**

for example:

```
ChildProject\Screen 1
ChildProject\VAR00001
ChildProject\Basic Script 1
ChildProject\Menu 1
```

The syntax for accessing the Parent project's resources from the Child Project is as follows:

**..\ResourceName**

for example:

..\Screen 1  
..\VAR00001  
..\Basic Script 1  
..\Menu 1

It must be taken into account that there could be different nesting levels. For instance, If you have a "Father Project" with two "child projects, "Child Project 1" and "Child Project 2" you can access the "Child Project 2" resources from "Child Project 1" by using the following syntax:

**..\ChildProjectName\ResourceName**

For instance:

..\ChildProject2\Screen 1  
..\ChildProject2\VAR00001  
..\ChildProject2\Basic Script 1  
..\ChildProject2\Menu 1

The other possibility would be to have a father project and a child project which itself has a further child project. Therefore you would have a "Father Project", a "Child Project" and a "Child Project2". In this case the syntax for accessing the "Child Project 2's resources" from the "Father Project" would be:

**ChildProjectName\ChildProjectName2\ResourceName**

for instance:

ChildProject\ChildProject2\Screen 1  
ChildProject\ChildProject2\VAR00001  
ChildProject\ChildProject2\Basic Script 1  
ChildProject\ChildProject2\Menu 1

The syntax for accessing the resources of the "Father Project" from the "Child Project2" would be:

**..\..\ResourceName**

for instance:

..\..\Screen 1  
..\..\VAR00001  
..\..\Basic Script 1  
..\..\Menu 1



Caution! If local screen variables have been set in the screen you are working in, the syntax to access the parent project variables may be different. When the parent project variable has the same name as the local variable you must use the following syntax:

..\..\<variable name>

The suffix"..\\" is used for returning back one hierarchy on the variable's access. The hierarchy is:

Local Screen Variables -> Project Variables -> Parent Project Variables

## Database Viewer

---

Data from the Child Project's Database can be displayed by the Parent Project. By using the execution properties of the **"Log Window"**, **"DataLogger Window"** and **"TraceDB Window"** objects you can select the 'Child Project' name to be linked in order to view its data.

## Strings in Child Projects

---

Each Child Project could have his activation language context and his local languages. The Child Project first will find the String ID in the local string and if it doesn't exist then the string will be found in the Father Project String Table. In this way it is not necessary to copy the Child Project strings in the Father Project strings.

## Child Project Users Inclusion

---

The User and Password management internal 'Child Projects' depends exclusively on settings carried out in the 'Parent Project'. When the Password management is enabled in the 'Parent Project' this means that it will also be active in 'Child Project'. On the other hand, when it is not active in the Parent Project it won't be active in the 'Child Project'. When the Password management is active, the 'Parent Project' users will be enabled, and therefore can also access the project commands of the 'Child Project' if they have requested rights to do so. The 'Child Project' Users are only active when they have been enabled in the settings of the Parent Project's **"Include Child Project Users"** group in the **"Users and User Groups Child Project Option Properties"**. In addition to this the Child Project User Max Level can also be specified for which the users must have in order to log on. Users with higher levels will not be acknowledged and denied access.

## 28. Networking

***The Networking exploits the potentialities of protocols capable of going on Ethernet networks for interlinking Movicon workstations in Client/Server mode.***

It is more than often that the distributed architecture of process control require plants to include multiple workstations, control or display stations which are used through one or two PCs linked up together in Ethernet network. Today's Ethernet network is the most used means of communication due to its hi-performances/low cost ratio.

This means, however, that more tools are needed for interlinking more stations capable of sharing process data.

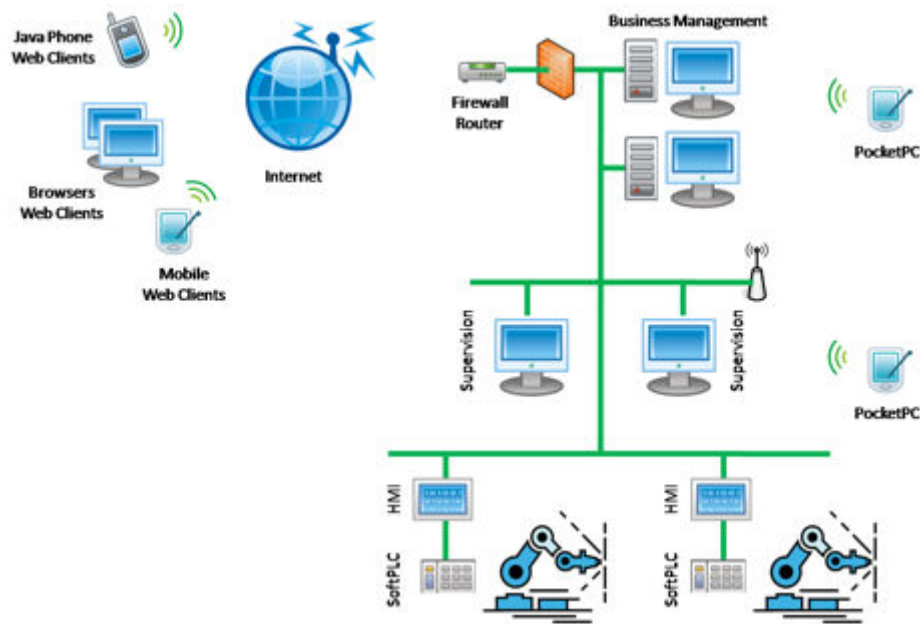
Movicon nevertheless is able to satisfy these Networking connectivity requirements by exploiting the TCP/IP, UDP, and HTTP protocols which are also used on the Ethernet network as being the most widely used and fastest.

Two or more Movicon stations can be interlinked with just a few simple mouse clicks.



The remote link functions (Networking) are enabled in Runtime only when the appropriate option on each hardware key of each PC station (Client or Server) is active.

The variables in use in a Client and exchanged with the server will naturally be in use in the server as well.



An example of the network architecture. Each PC station is connect in LAN network in Client/Server architect. Each Movicon station participant can read or write data to each other.

## Network Installation

Before going ahead with connecting two Movicon stations in network you need to provide connections of two PCs through a Ethernet network and install the TCP/IP protocol on the Microsoft Windows operating systems.

We will describe the procedures needed for connecting up two Movicon stations to each other by using a normal compatible Ethernet NE2000 card and the TCP/IP network protocol, which is presented in the Windows operating system, as an example.

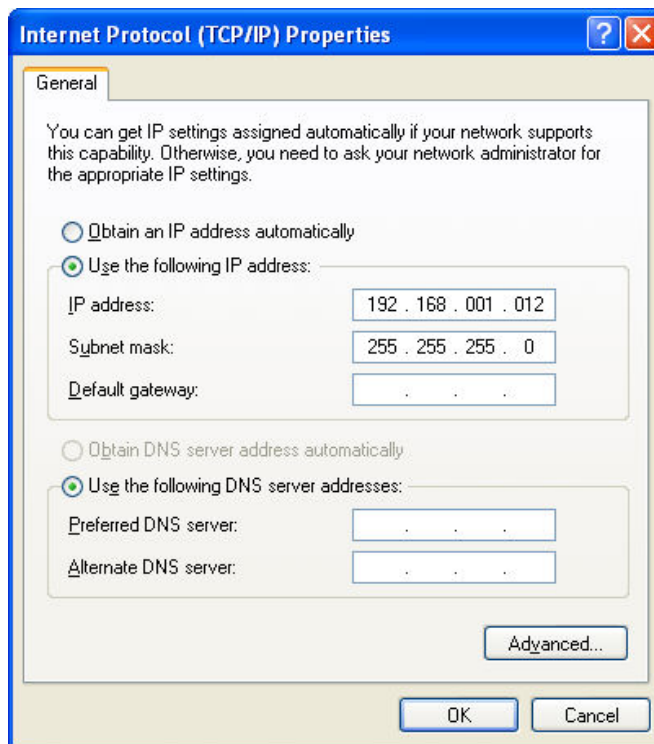


For any other network configuration permitting the use of the TCP/IP protocol please refer to its instructions.

After having installed the network card in the PC bus and having provided the relative configuration when a Plug & Play card is not being used, it is possible to use the Network functions from the Windows Control Panel to install the card and protocol in the system. When having activated the network functions from the control panel you need to proceed with installing the card and protocol by using the Add button and selecting the item desired:

We will select the card type, the makers, any drivers needed etc for the card.

As for the protocol, in our example we will install the Microsoft TCP/IP protocol and configure it by assigning a specific IP address, such as illustrated in the figure below:



Obviously each PC should receive in association its own IP address which is expressed as being unique by the last two figures.

A check should be made at the re-startup of the system to see if the network has been connected correctly and the protocol for the operating system is working properly.

In order to be completely sure that the TCP/IP network is working correctly you can use the PING.EXE utility provided in all Microsoft systems.

To carry out a check use the Run command from the Start menu or start the DOS prompt, then write the PING command followed by the IP address of the station whose connections you wish to check.

For instance, let's suppose we need to check the correct running of the communication towards the PC connect in net whose IP address is 198.162.200.14, therefore we shall write:

```
PING 192.168.200.14
<ENTER>
```

DOS window responding to the call made should appear showing the connection which was carried out:



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:

Reply from 192.168.0.1: bytes=32 time<1ms TTL=254
Reply from 192.168.0.1: bytes=32 time<1ms TTL=254
Reply from 192.168.0.1: bytes=32 time<1ms TTL=254
Reply from 192.168.0.1: bytes=32 time<1ms TTL=254

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\Administrator>_
```

Only in this way can we be certain that the network has been installed correctly to be able to connect data of the two Movicon stations.



Using WinXP SP2, remember to check the Access Rights for the Firewall.

## Network Connection

---

After installing the network correctly you can proceed with connecting data from one Movicon station to another.

Movicon permits one, a group or all the Real Time DB variables to be connected to the project through as many Real Time DB variables of another Movicon applied project existing on the same network.

To configure the connection of variables of one project to the variables of another project, proceed with using the **"Variable Network Client Properties"** through the 'Properties Window' to configure the necessary variables.

When you wish to connect a group of variables (or all the Database) you need to selection the ones needed and modify them through the **"Variable Network Client Properties"** from the 'Propertied Window'.

## Windows Terminal Server

---

The new Movicon Networking "Local Transport" management now allows easier project realization in the Terminal Server environment. The Movicon Terminal Server instances and projects started up as Service can communicate locally using the Networking and not the OPC (as was the case in the past).

To run Movicon from a Windows Terminal Server you need to use a certain configuration because each Terminal Server client runs a different Movicon project session. The system configuration must be done as follows:

1. The first Movicon project must be run as Service and has to manage field communications. It must be a Network Server and all the its project variables must be exposed to the Clients.
2. The second project must be a Network Client of the previous project, it must enable the "Local Transport" and it mustn't enable any communication driver.

When the Client project is run, it will acquire the variables from the Network Server project.

3. The Terminal Server session must run the Network Client project and NOT the Network Server project, which is however managed as service.
4. The Movicon license must be NET type (network license) to enable a number of users equal to the maximum number of Terminal Server connections desired at the same time.

The use of the Network Server-Client configuration is necessary because only one project must execute the communication drivers. If the Terminal Server runs a new project session which has been enabled with drivers there will be communication conflicts. Therefore the project, run from the Terminal Server, must not have its communication drivers enabled and thus must retrieve data from the field by exchanging variables through Networking from the Server project.

You must take into account that Log files will be created independently from both the Server and Client projects. Therefore it will be necessary that the two ODBC connections are different (if the two project names are different so will the ones for default be for the ODBC connections). If you want to display the Server project's Historical Log from the Client project you must use the grid object.

As regards to DataLoggers and Recipe with the need for one data source only, you will have to enable the registration in the Server project, set the Client project's ODBC connections to the Server project's database and disable the DataLogger/Recipe registration in the Client project.

## Remote Project Debugging

---

The remote project debug function allows you to connect to other Movicon instances being run on remote computers/devices (Win32 or WinCE). This makes it all possible to debug projects in remote and above all those projects being run on WinCE platforms. This function exploits a major part of the potentialities provided in the Movicon Networking.

The command to execute remote process debugs can be evoked in two modes:

1. by using the **"Attach to Process"** command from the ToolBar
2. by using the **"Attach to Process"** command from the Upload window in projects for WinCE

When executing the **"Attach to Process"** command a dialog window will open to insert the Server to be connected to. The Server can be indicated by its set IP address or its name, or by searching for it in the tree of connections available in the network to which the computer is connected.

After having set and confirmed the Server with OK, a second window will open requesting User authentication. The User name and password are obligatory only when the project being run has been enabled with the User management, otherwise click on the OK button straight away. When the project has been enabled with the User management, the user must have administrator level (level 1023) in order to be authenticated and granted connection. Furthermore the project must be enabled for remote debugging. This is done by enabling the "Debugger" option in the project's "Networking Server" properties (default = false).

After this procedure the project is started up in the same mode as the non startable child project. Therefore the variables in the controls are connected in dynamic mode ([\\NET]), the alarm, historic data viewer windows and the scheduler window are connected to the Networking Server. In addition to this all the project logic is not executed.

The remote debug provides the following functionalities:

- The Watch window allows the data relating to the Server project to be displayed and to monitor the variables in the Server
- The 'Scripts' new Watch window allows you to verify which Scripts have been loaded in memory and the total execution time of those in run mode. The debug window is opened by double clicking on a script
- A debug can be executed on the local and project IL logic (you can view the logic executed locally in the "Local IL Logic Window")

- The remote process can be restarted (not possible for the service). This possibility is proposed when disconnecting from the Server. A window opens where you can restart the project on the Server
- Analyses the project statistics with the right debug window (not supported on WinCE)

Other things to consider:

1. The remote debug messages have the lowest priority, they do not influence the normal running of the system where network traffic is concerned
2. The project to be connected to must be enabled with the "Networking Server" option (as well as the "Debugger" option) in the Networking Services settings
3. The Networking default transport is used for connecting in remote debug, this is the one selected in the "Client Network Settings" of the project to be connected to. This same transport must also be enabled in the Server's "Networking Service" otherwise any attempt to connect will fail
4. The screen's file path and name can be entered in the command line instead of those of the project and are exploited by Movicon to execute the remote project debugging

## Network Services Properties

---

The "Network Services" properties are used for defining the settings concerning both Server and Client projects. This particularly includes the possibility to select and configure which protocol to use for communicating through the Ethernet.

The available protocols are:

- **Local**
- **TCP**
- **UDP**
- **HTTP**



**Some of the advanced protocol options may slow down communications when dealing with big packets ("Sealing" and "Enable Encryption" properties), or with supplementary operations which the Server and Client must execute before sending packets ("Enable Compression" property).**

## Network Server Properties

---

The "Network Server" properties allow you to define the settings concerning the Server projects.

### Network Server

This property is used for enabling or disabling the Network Server for the project in question.

### Default Log On User

This property allows you to set the User that the Server project is to use for any anonymous remote connections. Therefore any anonymous Client trying to connect to the Network Server will be authenticated with the credentials of the user entered in this property. This option is only used when the user management is active in the Server project and behaves in the following way:

- A Web Client project will be automatically authenticated with the user set as "Default Log On User" in the Server. Therefore no login window will appear.



- A Network Client project without any user name entered for connecting to Server (Network Client Settings -> User), will be authenticated with the same user set as the "Default Log On User" in the Server. Therefore no "Network Server 'ServerName' Authentication failed" message will appear in the Client log window. Also "Alarm Windows" and "Scheduler Windows" will be authenticated using the user entered as the "Default Log On User" in the Server.

#### **Debugger**

This property is used for activating the debug function for the Networking communications.

#### **Max # Clients**

The maximum number of Client Stations, which can connect to the Server at the same time, is entered in this edit box.

#### **Max # Failing Actions**

The maximum number of error packets which can arrived from a Client Station before it put into quarantine is entered in this edit box.

#### **Network Server Threading Pool**

Number of threads to be used for handling Server packets. When inserting a number in the negative the subdivision of threads will be done based on the number of processors in the PC.

#### **Routing Table**

This property is used for re-addressing messages on another Server. In this case the PC works as a bridge. This maybe useful in situations where the destination Server is not visible from the source Server, for instance, when the destination Server is part of a local Lan and access is gained through a Server connected to internet.

## **Log and Security Properties**

---

The "Log and Security" properties are used for defining the settings for the Log files inherent to the network communications and any eventual filters on IP addresses which can access the Server.

#### **Log**

This property allows you to enable or disable the recording of Log files relating to the Networking communication information. The Log files are saved in the project's "NETLOG" folder in ASCII format and contain information relating to the connection status, variable changes, etc.

#### **Log File Size (Kb)**

This property allows you to set the maximum size (in Kb) for each Log file. When this size is reached the Log file will be closed and a new one will be created with a progressive number at the end.

#### **Max. Log File Age (days)**

This property allows you to set the maximum log file duration. When this limit has been reached, expressed in days, Movicon will start to recycle by overwriting the files starting with the oldest ones.

#### **Starting IP Address**

Movicon allows IP address ranges to be set for which the Server will accept connection requests. Addresses which are not in these ranges will be refused. In this property you can insert the start addresses of the range desired.

#### **Ending IP Address**

Movicon allows you to set IP address ranges for which the Server will accept connection requests. Addresses which are not in these ranges will be refused. In this property you can insert the ending addresses of the range desired.

#### **Clean Quarantine Address tick (min)**

This property allows you to set every how many minutes the IP addresses in quarantine must be cleaned. In this way any Client in quarantine can be accepted by the Server again if the request to do so does not continue to generate an error.

**Beep Under Attack**

When enabling this property, the Server will give out an acoustic signal every time an unauthorized Client tries to connect.

**Banned IP List**

This property allows you to set a list of IP addresses which will be banned from accessing the Server. A dialog window will open through which you can insert the list of IP addresses to be banned:

---

## Local Transport Properties

The Local Transport properties are used for defining the setting inherent to the Local communication protocol which concern the Network services. The "**LOCAL**" protocol is needed for communicating internal the same machine without using the net.

A network client project can connect to a local network server project. Any project not enabled as "Network Server" will startup its network services in client mode only if this is requested (it will not be committed to listen in on ports configured as Servers). This will allow the network client project's network connections to be set on "localhost" and to use the local or TCP transport for connecting to a Movicon network Server started up locally.



It would be best to leave the network client project with its local transport enabled, as this transport uses a shared memory block which is much more faster than the local network communication when using the TCP transport.

**Enable**

This property let's you enable or disable the use of the Local Transport for networking between Server-Client stations.

**Timeout (ms)**

This property allows to enter the timeout in milliseconds for the response of a message sent to a remote station. When this time expires a error message will be generated.

**Inactivity (ms)**

The set value (default 10 sec.) is used for keeping the resource active in the sockets so that to avoid having to repeatedly recreate it on the Server for the different Clients. In addition, this value is multiplied by 30 (default 10000 x 30 = 5 min) for managing the disconnection of inactive Clients. When a Client connection remains inactive for the time set here, it will be disconnected and reconnected only when the next request is made.

---

## TCP Transport Properties

The "TCP Transport" properties allow you to define the settings inherent to the TCP communication protocol concerning the Network services.

**Enable**

This property allows you to enable or disable the use of the "TCP Transport" between Server-Client stations.

**Timeout (ms)**

This property allows you to set the amount of time to wait for a response (expressed in ms) after a message has been sent to a remote station. When this time elapses an error message will be generated.

**Inactivity (ms)**

The set value (default 10 sec.) is used for keeping the resources active in the sockets as not to continually create on the Server for the various Clients. This value is multiplied by 30 (default 1000 x 30 = 5 mins) to manage the disconnection of inactive Clients. When a Client's connection is inactive for the time set, it will be disconnected and only reconnected on the next request.

**Port**

This property allows you to set the number of ports (sockets) to be used for the transport.

**Allow Routing**

This property allows you to enable the Transport to participate in routing messages addressed to other Transports.

## TCP Transport Advanced Settings Properties

---

The "Advanced Settings" properties allow you to define the settings inherent to TCP communication protocol regarding the Networking

**Transport Threading Pool**

The number of threads to be used for managing the processes linked to the Transport. When entering a negative number the sub-dividing of threads will be done based on the number of processors existing in the PC.

**Client Threads Limit**

This property is used for setting the number of threads to be entered at the same time for managing the Transport.

**Incoming Speed Limit**

This property is used for setting the maximum number of Bytes according to what can be received from the Transport.

**Outgoing Speed Limit**

This property is used for setting the maximum number of Bytes according to what can be transmitted by the Transport.

**Bind Source IP Address**

This property is used to ensure that Movicon opens the communicating port on the specified port. If nothing is specified in this field, the communication ports will be opened on the network card or ethernet communication channels predefined by Windows.

**Max. Message Size**

This property is used for entering the maximum message size which can be managed from the Transport.

**Nr. Attempts to Connect**

This property is used for setting the maximum number of connection attempts to a Server communication port before timeout expires. The default "0" value does not impose any limit on the number of connections attempts by the Client to a Server.

**Attempts Interval**

This property is used for setting the interval in milliseconds between one attempt to connect and the next. The default value is "1000".

**Enable Compression**

This property allows exchanged packets to be compressed. This is handy to use when transmitting with low band rates.

**Enable Encryption**

This property allows exchanged packets to be encrypted for higher security reasons. When publishing screen pages for the Web client the current TCP transport settings are used at that moment and set in the java applet parameters called in the html page published with IIS for the (Internet Information Services). However, these parameters can also be modified by opening the html page with an external editor. AES is used as the encryption algorithm with a 128 bit private encryption key property.



**This option must be enabled in both network Server and network Client in order for it to work properly.**

**Sealing**

This property is used for setting a integrity control on messages sent in networking. The TCP transport has its own integrity control on exchanged data. By setting a sealing type you will make data transmission and reception more secure, but slower. There are three modes to choose from:

- **None**
- **MD5**
- **CRC32**
- **Adler**

## TCP Transport SOCKS Options Properties

---

The "Socks Options" Properties allow you to set the settings inherent to the TCP communication protocol regarding the Networking services.

**Enable**

This property is used for enabling the Socks Option for the Transport in question.

**Server**

This property is used for specifying which is the Socks Server for the Transport.

**Port**

This property is used for setting the number of ports (sockets) to be used for the transport.

**User**

This property is used for specifying the name of the Socks user to be used.

**Version**

This property is used for selecting the Socks version to be used. The choices are:

- **Ver. 4**
- **Ver. 5**

**Enable Authentication**

This property is used for enabling the Socks authentication for the Transport.

**Authentication**

This property is used for selecting the authentication type to be executed. The choices are:

- **No Authentication**
- **User Code**

## UDP Transport Properties

---

The "UDP Transport" properties allow you to define the settings inherent to the UDP communication protocol concerning the Network services.

**Enable**

This property allows you to enable or disable the use of the "UDP Transport" between Server-Client stations.

**Port**

This property allows you to set the number of ports (sockets) to be used for the transport.

**Inactivity (ms)**

The set value (default 10 sec.) is used for keeping the resources active in the sockets as not to continually create on the Server for the various Clients. This value is multiplied by 30 (default 1000 x 30 = 5 mins) to manage the disconnection of inactive Clients. When a Client's connection is inactive for the time set, it will be disconnected and only reconnected on the next request.

#### **Allow Routing**

This property allows you to enable the Transport to participate in routing messages addressed to other Transports.

## **UDP Transport Advanced Settings Properties**

---

The "Advanced Settings" are used for defining the settings inherent to the UDP communication protocol concerning the Networking services.

#### **Bind Source IP Address**

This property is used for excluding or rendering the IP addresses, specified for the Transport in question, invisible.

#### **Max Message Size**

This property is used for inserting the maximum message size to be managed by the Transport.

#### **Enable Compression**

This property is used for compressing transmitted packets. This may be useful when the band rate used in the transmitter is low.

#### **Enable Encryption**

This property consents to crypting messages exchanged between network Client and Server.

When publishing screen pages for the Web client the current UDP transport settings are used at that moment and set in the java applet parameters called in the html page published with IIS for the (Internet Information Services). However, these parameters can also be modified by opening the html page with an external editor.

AES is used as the encryption algorithm with a 128 bit private encryption key property.



**This option must be enabled in both network Server and network Client in order for it to work properly.**

#### **Sealing**

This property is used for setting a integrity control on messages sent in networking. The UDP transport has its own integrity control on exchanged data. By setting a sealing type you will make data transmission and reception more secure, but slower. There are three modes to choose from:

- **None**
- **MD5**
- **CRC32**
- **Adler**

## **HTTP Transport property**

---

The "HTTP Transport" properties allow you to define the settings inherent to the HTTP communication protocol concerning the Networking services.

#### **Enable**

This property allows you to enable or disable the use of the "HTTP Transport" between Server-Client stations.

**Timeout (ms)**

This property allows you to set the amount of time to wait for a response (expressed in ms) after a message has been sent to a remote station. When this time elapses an error message will be generated.

**Inactivity (ms)**

The set value (default 10 sec.) is used for keeping the resources active in the sockets as not to continually create on the Server for the various Clients. This value is multiplied by 30 (default 1000 x 30 = 5 mins) to manage the disconnection of inactive Clients. When a Client's connection is inactive for the time set, it will be disconnected and only reconnected on the next request.

**Port**

This property allows you to set the number of ports (sockets) to be used for the transport.

**Allow Routing**

This property allows you to enable the Transport to participate in routing messaged addressed to other Transports.

## HTTP Transport Advanced Settings Properties

---

The "Advanced Settings" properties allow you to define the settings inherent to HTTP communication protocol regarding the Networking services.

**Transport Threading Pool**

The number of threads to be used for managing the processes linked to the Transport. When entering a negative number the sub-dividing of threads will be done based on the number of processors existing in the PC.

**Client Threads Limit**

This property is used for setting the number of threads to be entered at the same time for managing the Transport.

**Incoming Speed Limit**

This property is used for setting the maximum number of Bytes according to what can be received from the Transport.

**Outgoing Speed Limit**

This property is used for setting the maximum number of Bytes according to what can be transmitted by the Transport.

**Bind Source IP Address**

This property is used for excluding or rendering the IP address, specified to the Transport in question, invisible.

**Max. Message Size**

This property is used for entering the maximum message size which can be managed from the Transport.

**Nr. Attempts to Connect**

This property allows you set the max. number of attempts to connect to a Server communication port before timeout expires. The default value '0' does not limit the number of attempts by the client to connect to the Server.

**Attempts Interval**

This property allows you to set an interval in milliseconds between a connection attempt and the next. The default value is "1000".

**Enable Compression**

This property allows exchanged packets to be compressed. This is handy to use when transmitting with low band rates.

**Enable Encryption**

This property allows exchanged packets to be encrypted for higher security reasons.

When publishing screen pages for the Web client the current HTTP transport settings are used at that moment and set in the java applet parameters called in the html page published with IIS for the (Internet Information Services). However, these parameters can also be modified by opening the html page with an external editor. AES is used as the encryption algorithm with a 128 bit private encryption key property.



**This option must be enabled in both network Server and network Client in order for it to work properly.**

#### **Sealing**

This property is used for setting a integrity control on messages sent in networking. The HTTP transport has its own integrity control on exchanged data. By setting a sealing type you will make data transmission and reception more secure, but slower. There are three modes to choose from:

- **None**
- **MD5**
- **CRC32**
- **Adler**

### **HTTP Transport SOCKS Options Properties**

---

The "Socks Options" Properties allow you to set the settings inherent to the HTTP communication protocol regarding the Networking services.

#### **Enable**

This property is used for enabling the Socks Option for the Transport in question.

#### **Server**

This property is used for specifying which is the Socks Server for the Transport.

#### **Port**

This property is used for setting the number of ports (sockets) to be used for the transport.

#### **User**

This property is used for specifying the name of the Socks user to be used.

#### **Version**

This property is used for selecting the Socks version to be used. The choices are:

- **Ver. 4**
- **Ver. 5**

#### **Enable Authentication**

This property is used for enabling the Socks authentication for the Transport.

#### **Authentication**

This property is used for selecting the authentication type to be executed. The choices are:

- **No Authentication**
- **User Code**

#### **Enable HTTP Proxy Server**

This property enables the use of the Proxy Server to establish the connection to the Server.

#### **HTTP Proxy Server**

This property is used for setting the Proxy Server's IP address.

#### **HTTP Proxy Server Port**

This property is used for setting the Proxy Server's port.

# Setting up Network Clients

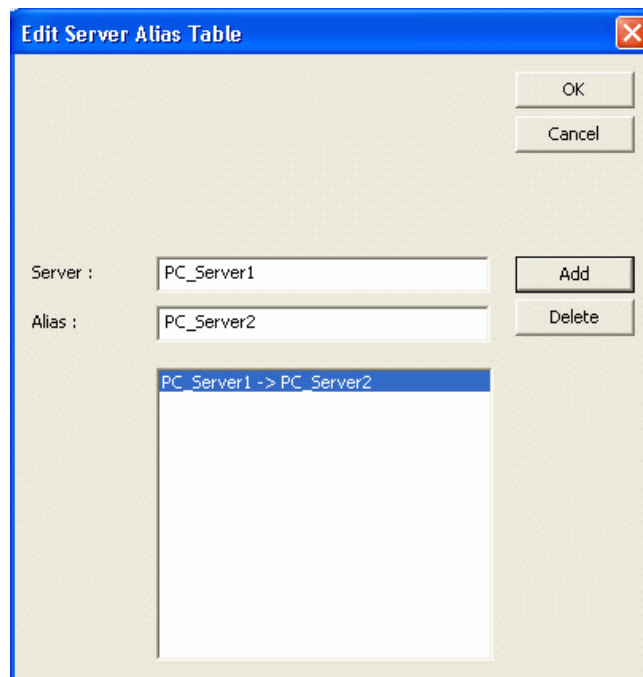
The 'Network Client' properties allow the settings concerning Client projects to be defined. The Client settings can be personalized in a particular way according to the Server which must be connected. Therefore 'Client Profiles' can be inserted ("**Add new Client Rules**" command) to diversify the Client's properties according to the Server to be connected (for example a different protocol may be selected for each Server). In addition to this you can also add 'RAS Stations' ("**Add new RAS Station**" command) in order to connect to a Server through phone connections.

## Network Client Properties

The "Network Client" properties allow you to define the settings concerning Client projects.

### Server Alias Table

This property allows you to re-route Server names with other names. This function is very handy, for instance, when having to change names of computers belonging to the network where this operation can be done altogether in one point of the project. The dialog window for editing the table of Aliases is as shown below:



The fields shown in the Alias window have the following meanings:

- **Server:** Name of the Server to which the project has to connect to when the Alias Table is not enabled. This is the name which the Server was set with in the "**Network Server**" property of the "**Variable Network Client Properties**"
- **Alias:** Name of the Server in which the connections must be re-addressed.



Using the Alias Table is very handy also in the designing and debug phases. You can, in fact, quickly re-route the Server's name in project tests. In addition, by knowing the plant server's name in advance you can insert it in the project and use it's Alias for internal testing in the office.



**Black List Expiry Time**

This property consents you to customize the Server expiry time on the Black List. The time is expressed in milliseconds and the default value is equal to 60,000. A Client can put the IP or the name of a Network Server on its Black List in cases where a timeout error has occurred for that Server. The Server will then remain on the Black List for the time set in this property and the Client will not make any attempts to connect to that Server during this time. The Black List management consents faster re-connections of variables from the primary Server to the backup Server, or viceversa. In Client-Server network configurations with only one network Server, it may be convenient to set this time to zero in order to eliminate any waiting times in reconnecting the Server which went into timeout.

## Network Client General Properties

---

The "General" properties are used for defining time and transport settings to be used for Client projects.

**Timeout (ms)**

The Server's answer timeout (expressed in ms) after a message has been sent from the Client is entered in this property. An error message will be generated when the Timeout has expired.

**User**

This property is used for entering the name of the user with which the Client will represent itself to the Server. This setting is only significant when the Server project has the 'Password Management' enabled. In this case the user must also be present in the Server project to be acknowledged. By doing this the Client acquired the rights associated to the user in question, and can get access to variables based on these rights.

**Ping Time (ms)**

This property is used for setting the ping time to be used while connecting to the Server (the "0" value avoids the use of the ping time).

**Transport**

This property is used for selecting the Transport (Protocol) type to be used by the Client for communicating with the Server.

**Priority**

This property is used for associating a priority level to the connection in question. The values are from 0 to 100. The higher the number, the higher the priority therefore the highest priority is a 100.

**RAS Station**

This property is used for selecting a RAS connection (which must be created beforehand) to connect to Server exploiting telephone lines.

**Enable RAS Station**

This property is used for enabling the use of a RAS station to connect to the Server exploiting telephones lines.

## OPC XML DA Client Properties

---

The "OPC XML DA Client" properties allow you to define the settings concerning the Movicon XML Client.

**Thread Pool**

The number of threads to be used for requests to the XML Server. If you have to connect to more than one Server you can enter more threads, one for each Server. The value -1 is the setting for default.

**Poll Time Server In Use (ms)**

This edit box is used for setting the polling time in milliseconds towards the Server. This setting can be changed in Runtime with the appropriate Basic Script functions.

**Poll Time Server NOT In User (ms)**

This edit box allows you to set the polling time towards the Server in milliseconds when the variable are not in use. This setting can be changed in Runtime with the appropriate Basic Script functions.

**Write Behind Delay (ms)**

This edit box allows you to set the time in milliseconds for the Write Behind Delay. The OPC XML DA Client manager will delay the writing towards the Server according to the time set in this box.

## RAS Stations

---

The Network communication between two Movicon stations can be done through a Ethernet network and through a RAS connection via modem as well. To make this possible you need to create an "Incoming Connection" on the PC Server, so that when the Client makes a call and connection is established, the two PCs will be linked as if they were networking with each other.

To add "RAS Stations" you need to use the **"Add new RAS Station"** command.



The **RAS Connection** is a Operating System function that allows a Server-Client type connection between two stations using a connection via Modem. Once the connection has been established the TCP/IP protocol can be used for exchanging data between Server and Client. The RAS connection can be created from the "Control Panel - Network Connections". However, the configuration procedure may change slightly according to the Operating System being used (ie. Windows XP/Vista).

The RAS Station must be created only on the PC acting as Client, while an "RAS Incoming Connection" must be configured on the PC Server which will allow it to respond to the Client's calls automatically.

For further information on how to set up a "RAS Connection" please consult the relevant form on the "Web Support Center".



Be careful not to confuse a **"RAS Station"** with a **"RAS Connection"**. The "RAS Station" is in fact a component of the Movicon project, which is used for setting up communication parameters between Server and Client. The "RAS Connection", however, is a Operating System Setup which is used for establishing a physical connection between Server and Client.

## RAS Stations General Properties

---

The 'General' properties allow you to define the RAS Station settings for the Client project.

**Name**

The name of the RAS station is entered in this property. The Name is essential for being able to identify and access the properties and methods of the RAS Station through the Basic Script functions.

**Dial-up**

The name of the RAS Connection to be used for connecting the Client to the Server is entered in this property. In this case the RAS Connection should be created and configured in the Operating System beforehand. When this field is left empty you will need to fill in the next fields: "Telephone Number", "User Name" and "Password".

For further information on how to configure a "RAS Connection" please consult the appropriate card available on the "Web Support Center".

**Phone Number**

The telephone number which the Client station must dial is entered in this field. The number should correspond to the line connected to the Server station.  
If the "Connection" property has been filled in this field my be left empty.

**User Name**

The user's name with which the Client station is to be authenticated by the Server station is entered in this property.  
In this case the user must be a user known to the Operating system and therefore not necessarily a user of the Project Server, but a user declared in the Operating System.  
If the "Connection" property has been filled in, this field my be left empty.

**Password**

The user's password with which the Client station is to be authenticated by the Server station is entered in this property.  
If the "Connection" property has been filled in this field my be left empty.

**Retries**

The maximum number of connection attempts, which are to be carried out when the called fails first time, is entered in this property.

**Disconnect after (sec)**

How long after the inactivity time is the connection to be closed is entered in this property.  
The time count starts the moment in which all the variables connected to the Server are no longer in use.

**Retry Hold Time (sec)**

The Retry Hold Time, before re-dialing when connection fails, is entered in this property.

**Prompt Before Connecting**

When enabling this property a confirmation window will appear each time Movicon execute a connection. In this case the operator must confirm or abort the call.

**Show Dlg**

When enabling this property a window will appear during the calling phase showing the current status of the connection.

## Client Rules

---

The Movicon Networking communication architecture allows two or more stations to connect to each other with the possibility for additional Servers and Clients. When a Client station has to connect to a Server only the settings need configuring through the **"Network Client General Properties"**. But when a Client station must connect to two or more Servers you will need to configure various communication setups between the Client and the various Servers (eg. communication protocols, eventual RAS Stations, etc.). In this situation it would be appropriate to create **'Client Rules'** to customize the settings of the Client according to the Server to be connected to. By doing this the Client will be able to connect to the Server by using the **'Client Rules'** settings and not the general settings specified in the **"Network Client General Properties"**.

The **"Add new Client Rules"** command is used for adding 'Client Rules'.



The name of the 'Client Rules' must be the same as the Server to which the rules refer to. When a IP address is specified, the name of the "Client Rules" should be the same as the IP address.

## Client Rules General Properties

---

The 'General' properties allow you to define the 'Client Rules' settings for Client Projects.

Apart from the 'Name' property all the others are the same as already described in **"Network Client General Properties"**.

### **Name**

This property is used for entering the name of the Client Rules.



**The 'Client Rules' name must be the same as the Server's name to which the rules refer to. When a Client project connects to a certain Server it verifies whether one exists with the same Server name on the Client Rules list. In this case the connection is carried out according to the settings specified in the Client Rules, otherwise the Client's general settings will be used.**

**When a IP address is specified, the name of the "Client Rules" should be the same as the IP address.**

The name is also essential for being able to identify and access properties and methods of the Client Rules through the Basic Script functions.



## 29. Redundancy

---

***Movicon embeds powerful automatic functions to support the Hot Backup of critical stations workstations redundantly connected in network. Slave intervention, re-entry, data synchronization are managed in completely in automatic.***

Some critical processes under supervision and control require emergency station intervention, known as Secondary Server, when the main PC unit, known as Primary Server, crashes.

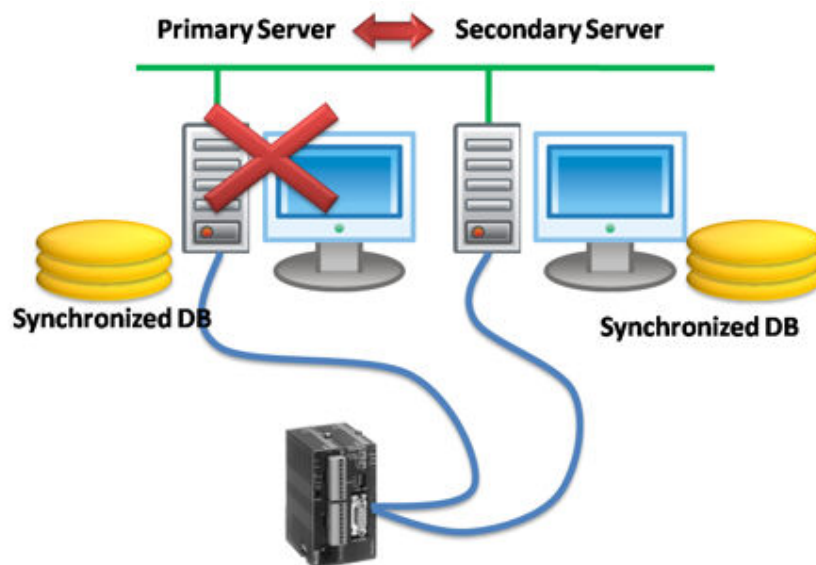
This ensures that the process's supervision and control service will continue to function when the Master supervision station suddenly becomes unavailable or crashes.



The redundancy management, in a supervision and control system, has the task of preventing data and control functionality of the plant.



**The redundancy functions are enabled in runtime only when the Redundancy and Networking option in each hardware key of every PC station (Primary or Secondary) have been enabled.**



An example of a redundancy architecture for supervision stations. The Secondary Server enters into function automatically when the Primary PC crashes.

## Redundancy Levels

---

The Redundancy's role in automation systems is to completely take over and substitute the primary component when it crashes with all the identical functionalities until the primary component has been repaired and put back into action.

The "Hot Backup" provides a secondary server to go into action automatically without requiring any manual intervention from the operator.

The redundancy concept can be applied to both software and hardware to stop the system losing any data or functionalities when transferring control over to the secondary backup component from the Primary component.

The redundancy concept can be applied to the following components in automation systems:

**PLC**  
    **CPU**  
    **I/O Cards**

**Connections**  
    **Bus**  
    **Network Cable**  
    **Serial**

**Computer**  
    **Display System**  
    **Control System**

The Redundancy functions embedded in Movicon support the Backup functionalities in your computer, allowing all the communication, display and control functions from the Primary PC Server to be transferred to a Secondary Server entirely in automatic.

This proprietary technology adopted by Movicon to perform these functions permits very fast synchronization times of up to less than a second when great amounts of data are involved. This is due to synchronizing data acquired during the emergency take over period by transmitting data in binary format instead of database format structures.

## Redundancy Functionality

---

The redundancy management, whether with Master functionality or Slave functionality, is completely embedded in the Movicon software and ensures secondary system intervention in complete automatic mode after primary system timeout set up in the system's configuration.

The redundancy functions are imbedded and native in the following critical system functionalities in Movicon:

- **Communication Drivers management**
- **Data Logger management**
- **Historical Log management**
- **Alarm management**

Any one of these functionalities is kept on Stand-by in the secondary station, and becomes active when the Primary station crashes to safeguard against losing control over the plant. The Movicon redundancy control system requires two PC stations connected in network based on TCP/IP, with Primary and Secondary Server functions:

- **Primary Server:** is the workstation which is conditioned to function as normal to manage the plant, by communication with it, acquiring data and providing control. If this station crashes, the secondary station will enter into function and assume complete control
- **Secondary Server:** is the workstation which is conditioned to function as normal to manage the plant in redundancy mode, through shared variable memory areas. This station allows plant interaction in independent mode and supplies all the same identical functions of the primary station. If the Primary server crashes, the secondary server will automatically assume plant management by starting up

the communication driver and recording engine functions, acquiring data and performing process control

### **"Normal" Conditions**

In normal working conditions, both the Primary and Secondary Servers are operative in the plant according to distinguished functionalities.

The Primary Server is dedicated to manage the driver's communication and historically log data on hard disk, according to the normal functioning of each Movicon application.

The Secondary Server is kept on stand-by and operative to carry out the same functions as the Primary in independent mode, except for the following different operating modes:

- The drivers of the secondary server are put on Stand-by and do not communicate directly. Operability of the secondary server is based on mirroring the variable memory areas, which are shared in automatic and transparent mode on the Primary server. As a consequence a command towards the field can be performed both on the Primary or Secondary indifferently, but change page performances must be done locally as each workstation processes its own graphic functions locally
- The Secondary's archives (Data Logger and Historical Log) do not work directly, to ensure absolute identical recorded data. The appropriated system redundancy functions make sure that data acquired and recorded by the primary are archived identically and directly on the secondary. This ensures complete data integration with time precision
- The cache memory shows the alarm situation of the primary's alarms and updates by mirroring the primary's cache for the same reasons as above
- The execution of the IL Logic associated to the Secondary will be put in pause mode
- the Event list and the Events in script variables will not be executed in the Secondary Server

### **"Emergency" Conditions**

When the primary server stops working (crashed, error, hardware damages), the secondary server will go into action immediately and communicate with the driver on stand-by and the execution all of the project's logic and Events is reactivated. The configurable response time is immediate to the second.

The historical engines will start to record the plant data directly the moment it enters into service to assess how much data has to be sent to the primary server the moment it returns into action. This will allow you to optimize the historicals' synchronizing times.

Once the Primary server re-enters into action it will automatically synchronize the historicals' status and the alarm situation in order to completely restore its functionality and archives to avoid any data loss.

The system, by exploiting its own technology, will only send the data recorded during the emergency period to the primary server by transferring the data in binary mode without using any database structures.

Once synchronization has been completed, which is automatically done by the redundancy management embedded in the system, the secondary server will return on Stand-by.



For correct use of the redundancy management you must install and configure the network form and the TCP/IP protocol in the operating system being used beforehand. For more information on how to do this please refer to the section on Networking.

### **Data Synchronization**

The redundancy management optimizes those projects which have many Data Loggers and a high sampling frequency because they have to manage large volumes of historical data. This management mainly limits memory used for data synchronization files, to avoid any system memory saturation in cases where large amounts data MBytes have to be synchronized. Two properties have been provided in the redundancy to deal with this allowing data synchronization to be managed with:

#### **Max Historical Cache**

#### **Max. Nr. Historical Cache Files**

The data synchronization files are created in the project's "DATA" folder. This files signify:



File Extension	Description	Struttura XML
.dmr	Cache file for the project's Data Logger.	<pre>&lt;?xml version="1.0" encoding="ISO-8859-1" ?&gt; /el/e Node with all the necessary information for recording data records of a Data Logger /el/e/HE Node with the information relating to the Data Logger that has generated a record: Data Logger name, date and time of recording and standard column values /el/e/list Node with the information on the recorded record columns /el/e/list/c Node with the information on one of the recorded record columns /el/e/list/c/v Node with the value information of one of the recorded record.</pre>
.hlr	Cache file for the project's Historical Log tables.	<pre>&lt;?xml version="1.0" encoding="ISO-8859-1" ?&gt; /el/e Node with all the information needed for recording a Historical Log table's data record. /el/e/HE Node with all the information relating to the record to be recorded: table name, data and time and standard column values.</pre>
.tbr	Cache file for the Trace DB tables for the Project's variables.	<pre>&lt;?xml version="1.0" encoding="ISO-8859-1" ?&gt; /el/e Node with all the information needed for recording a data record of a variable's Trace DB table. /el/e/HE Node with the information relating to the variable which has generated a Trace DB record: table name, date and time of recording and standard column values. /el/e/vb Node with information of the variable's previous value. /el/e/va Node with information on the current variable's value.</pre>

## Advanced Redundancy Functions

Movicon has some special functions embedded in the critical system redundancy management to extend the functions provided for the user.

## Synchronized Time Frequency

The Movicon Redundancy has an automatic tool embedded in the complex synchronization control management between two PC Servers: the automatic synchronization of the system's clock on the Secondary Server by the Primary Server.

Automatic Secondary Server clock synchronization always ensures the user that data is always congruent in all circumstances.

Synchronized time frequency takes place according to how the **"Sych. Time Frequency"** property from the "Redundancy Settings" has been set.

## System Diagnostics

Movicon gives out information system messages when the Secondary Server has entered into action, the re-entry of the Primary Server and the data being synchronized etc. This information will be displayed on both systems and recorded in the Historical Log automatically.

This will enable the operator to always have the information s/he needs to carry out an accurate analysis to value how the critical system is working.

## Status Variable

The redundancy management also provides the programmer the option to use a status variable, in order to inform the logic on the operating status of the Primary and Secondary Servers. For further information please refer to the "**Status Variable**" property from the "Redundancy Settings".

Some Basic Script functions dedicated to the Redundancy management have also been provided.

## Redundancy Restrictions

The Redundancy management used in a critical plant situation has some restrictions as regards to how some of the project's functionalities work and which the programmer should keep in mind.

This would be best done by considering those Movicon functions which are not critical and not managed in redundancy.



These functions must be evaluated carefully as their non redundancy use may effect the system backup management in a critical situation.

The functions not subject to Movicon project redundancy are:

Function	Description
System Variables	Movicon System Variables can not be made redundant.
Alarm Basic Scripts	The Alarm Basic Script interface is not managed in redundancy.
Trend Output	Out put on Trend files is not managed in redundancy due to the fact that the Trend's connection to the Data Logger is managed for this.

Furthermore, you must keep in mind that while the IL Logic associated to the project is executed only on the active Server, the screen or designs' Local IL Logic is always executed on both the Primary and Secondary Servers independently from which one is active. To avoid any problems you need to manage the Local IL Logic shrewdly by using, for example, the screens' local variables.

When using the communication driver's basic script functions you will need to introduce the right controls in the codes. In this case, for example, before executing the functions for creating and the task management, it will be necessary to verify whether the station is communicating (verify whether there are any errors and its quality status). Otherwise, the driver may not be loaded because being used by another Server. You can also refer to the appropriate status variables to see if the Secondary Server has driver control.



Caution: The Redundancy function and the data synchronization mechanism may present criticalities when the total amount of data to be managed is very high. Therefore, it is advised you carefully read the section on "Data Synchronization" and to also consider the hardware characteristics of the PC being used.

## System Requirements

---

A redundant system must be based on a configuration which allows the network connection of two PCs, which do not have to be necessarily identical to the hardware's configuration.

Both systems should at least give the same performances and have the same memory capacity while having identical redundancy functionalities.

It is fundamental that both systems are correctly installed with the network card and TC/IP protocol, on which the whole redundancy management is based.

## Redundancy Settings

---

The Redundancy Settings can be configured through the "Network Services" resource found in the "Project Explorer" window. To edit the Redundancy settings, select the group with the mouse and use the Movicon **"Properties Window"**:

### Type

This edit box is used for selecting the Server type to be used for running the project. The options are:

- **None**
- **Primary**
- **Secondary**

When the "None" option is selected the Redundancy functionality will not be activated.

### Server

The name or the IP address of the Primary Server is entered here. This property is set only in the Secondary Server project.

The IP address or Server name which is to be set in this redundancy field is also used as Backup Network Server for each variable that is not set with a Backup Network Server.

### Backup Server

The name or IP address of the Primary Server is entered in this box. This property is set only in Secondary Server projects.

The Secondary Server will attempt to connect to the Primary backup Server when the main Primary Server is not available. Once the connection remains connected to the backup Server, and failure to do so, an attempt will be made to connect to the main Primary Server. Failure of both connection procedures to one of the two primary Servers will activate the Secondary Server. When the Backup Server is not specified only the main Primary Server will be used. The 6 bit in the status variable indicates that the primary server is connected through the Backup Server. In cases where the Secondary is the active Server, the 6 bit will blink each time the secondary controls the presence of the backup server.

### Timeout

This edit box is used for entering the timeout after which the Secondary Server will take over control when the Primary Server crashes, disconnected etc.

### Retries

The number of connection retries the Secondary Server should carry out before going into action is entered in this box.

### Status Variable

This box is used for selecting one of the variables from the Movicon Real Time DB which will be used as the Redundancy's status variable. The meaning of each bit of the selected variable depends on the Server where the project is being run:

- **Bit 0:** Primary Server. This bit is set at true only in the project set as Primary Server
- **Bit 1:** Secondary Server. This bit is set at true only in the project set as Secondary Server

- **Bit 2:** Active Server. This bit is set at true only in the Server project active at that moment
- **Bit 3:** Secondary Connected. This bit will be set at true only in Primary Server projects when the Secondary project is connect to the Primary Server.
- **Bit 4:** Synchronization in course. This bit is set at true in both projects, Primary and Secondary, when synchronization is taking place between the Secondary Server and Primary Server
- **Bit 5:** Drivers Started Up on Secondary. This bit is set at true only in the project set as Secondary Server when its Communication Drivers start working due to an error in the Primary Server, eg. crash, disconnected etc
- **Bit 6:** Secondary Server. This bit will be true when the Secondary Server is connected to the Primary Backup Server. It will start blinking when controlling the presence of the Secondary Backup Server
- **Bit 7:** not used

**Switch on Driver Error**

This selection box is used for activating the Secondary Server to switch on its own Communication Drivers when any errors in the Primary Server's Communication Drivers occur. This switching on can also happen without the Secondary Server taking over plant control.

**Driver Error Timeout**

This box is used for entering the timeout after which the Secondary Server will switch on its own Communication Drivers in event of problems in the Primary Server's Communication Drivers.

**Max Historical Cache**

This box is used for setting the maximum number of records in the historical cache file for redundancy (default value = 100). A new cache file will be created when this value is exceeded. Historical data is cached when the primary or secondary project is verified to be the only when active. The cache files are then used for synchronizing the two primary and secondary projects.

**Max. Nr. Historical Cache Files**

This box is used for entering the maximum number of files which can be used for recording historical data which must be made redundant. There cache files are created on disk and used as unvolatile memory for synchronizing historical data of the two primary and secondary project.

**Sych. Time Frequency**

The number or hours the Secondary Server is to synchronize its time frequency with that of the Primary Server is entered here. The synchronization will not be executed when the "0" value is entered.



## 30. OPC (Ole for Process Control)

---

***OPC is the most commonly used standard due to its user-friendly communication modalities based on different bus technologies.***

OPC stands for "OLE for Process Control", where OLE refers to the Windows' abbreviation of: "Object Linking and Embedding", a powerful automation interface for the Windows applications.

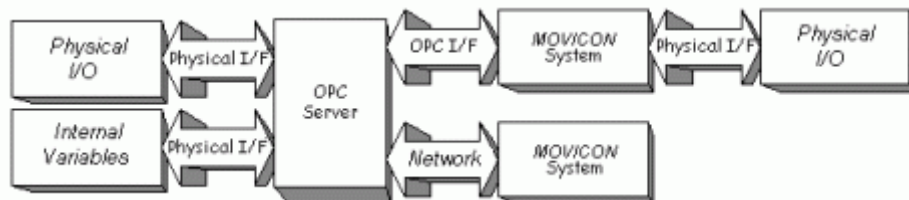
OPC simply rounds off the programming concept which implements an unified interface of different bus technologies on one part and programs for vision and automation on the other. Thanks to the standard technologies, today's communication between automation devices means more independence for the individual producer, especially when their apparatus supports the OPC standard. Thanks to the unified interface, hardware producers can always guarantee OPC interface availability by equipping their hardware product with OPC Server software products. This will ensure the user of being able to interface their purchased product with any applied software supporting the OPC standard.



The OPC standard is defined by public specifications released by the **OPC Foundation** consortium which are adhered to by Microsoft and all the major automation manufactures worldwide including Progea.

Movicon is an application based both on the OPC Client and OPC Server standards. As Client it can interface with all the OPC Serves with follow the 1.0A and 2.0 standards; as Server it allows all or part of the project variables to be shared with other applications and notifies them of events such as: Alarms, messages, system information and variable changes within the project.

In view of the increasing consolidation of the OPC standard in industrial automation, this technology must be considered as a communication mode with bus network devices or with other local or remote applications only.



This figure shows a block diagram of a typical OPC communication.

### OPC Client Editor

---

In the OPC communication Movicon has provided a Client interface which can be entirely configured and which supports any OPC Server created according to the OPC 1.0A and 2.0 specifications. This resource, integrated into the system, is identified with the "OPC Client DA (COM)" name and can be accessed through the Movicon "Project Explorer" window.

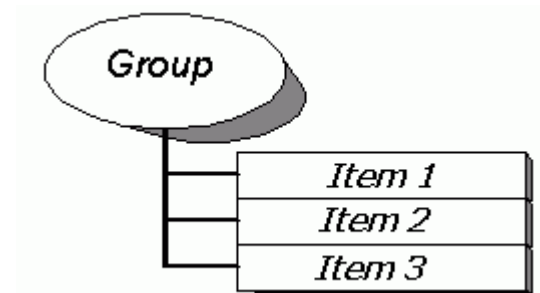
By using this resource you can setup the communication in OPC with one or more Servers. Movicon displays a list of Servers available on the local computer, but you can

also go and use OPC Servers installed on other remote computers connected in net or on the web.

Two OPC Server interfaces are currently supported: "DataAccess" and "Alarms&Events", identified respectively with the signs "DA" and "AE".

#### Data Access Specification

The first type of Server is organised in a structure composed of Groups and Items, as shown below, and permits I/O or variables in devices, which form part of the network field bus, to connect with variables from Movicon projects.



#### Alarms & Events Specifications

The second type of OPC Server, "Alarm&Event", notifies the occurrence of a specific event and alarm condition, configured within the Server itself, therefore it can be acknowledged with the Movicon project and managed accordingly.



The managing of events and alarms sent by the AE OPC Server can be done by using the corresponding event of any basic script within the project.

#### Importing from other Projects

---

Movicon allows any object (OPC Server, Groups, Items) setup within the "OPC Client DA (COM)" resource to be copied from one project to another. In order to carry out this procedure, you need to activate the project containing the parts to be copied, select the objects desired from the "OPC Client DA (COM)" resource, execute the Copy command, then activate the current project again, from the "OPC Client DA (COM)" local resource, and execute the Paste command. The copied objects will then be also made available in the current project.

The Drag & Drop technique can also be used to carry out this procedure.

#### DCOM Settings

---

The connection to a OPC network Server involves the usual necessity to setup the Windows DCOM services so that Clients are allowed access to the Servers. To get easy access to the DCOM configuration window use the "**Edit DCOM Settings**" command in the "Commands window" from the "Project Explorer" or from the text menu which appears by right mouse clicking on the name of the interested OPC.

#### OPC Client Log

---

the OPC client Log can be set for recording information relating to events received for value change notification items and items written to opc servers. This is made possible by setting the "OPCClient\Trace" key to value "1" (key is already documented in the list of registry keys).

#### OPC Client DA Properties

---

The OPC Client DA properties allow you to set some of the parameters for the dynamic Items and test functions.

To change the OPC Client DA properties, select the resource with the mouse and use the Movicon **"Properties Window"**.

#### **Default Dynamic Update Rate**

The update time for the project's dynamic OPC Items is entered in this edit box.

#### **Add Item Timeout**

The Timeout time, in milliseconds, for inserting a new dynamic OPC Item is entered in this edit box.

#### **Startup Timeout**

This property controls the timeout for initializing the OPC Server. When the OPC Server does not initialize within the time set, including all items set, a "Cannot start OPC Client Manager Thread" message will show. The default value is 30000 msec, but this value may not be enough for projects with very high numbers of items which are synchronized at startup and therefore will need to be increased.

This property can also be accessed through the "StartupTimeout" of the "OPCClientCmdTarget" basic script function.

#### **# Test Thread Pool**

This edit box is used for entering the number of Threads to be used for the OPC Item test run upon connecting to the Server.

#### **Test Item Timeout**

This edit box is used for entering the Timeout, in milliseconds, for the OPC Item test run upon connecting to the Server.

## **OPC Server DA Properties**

---

The OPC Server DA properties allow you to set some of the OPC Server parameters so that Movicon can connect to it afterwards.

To change the OPC Server DA properties, select the resource with the mouse and use the Movicon **"Properties Window"**.

#### **Name**

This edit box is used for entering the name of the OPC server. This is filled in automatically by Movicon when selected from the "Tag Browser" window.

#### **Network Node**

This edit box is used for entering the OPC Server's network path. This is filled in automatically by Movicon when selected from the "Tag Browser" window. The Local Server will be entered when left empty.

#### **CLSID**

The OPC Server's CLSID code is entered in this edit box.

#### **Reconnect time**

The time after which the Server will be reconnected is entered in this edit box in milliseconds.

## **Entering New OPC Servers**

---

To enter a new Server into the "OPC Client DA (COM)" resource is done by selecting the "Add a New OPC Item" command from the Project Explorer's Commands window or by selecting the same command from the menu which appears after double-clicking on the "OPC Client DA (COM)" resource.

In both cases a "Tag Browser" window will appear showing a list of available OPC Servers in the local computer and a list of Servers existing in other computers in the network.



In the window which shows the list of available Servers you can select a local Server from the "Local (My Computer)" group or a Server residing in another PC on the network from the "Microsoft Windows Network" group. Whereas the "Internet Servers (OPC XML DA)" group allows you to select a OPC XML Server instead.

#### Filter

In this text box you can execute filters for displaying the selected Server's items. The filter is then applied on the right side of the window and in order that the list is updated in accordance with the filter you need to execute the **"Refresh"** command with the appropriate button.

#### Server

The name of the Server with which you wish to connect to is entered in this text box. A further setup window is opened by using the Browse button on the right where you can enter the User and Password for Server access.

#### Live Data

When this option box is enabled the list of Items displayed on the right side of the window will be refreshed according to the values of the Server.

## OPC Item Connection String

---

In the project's Real Time DB XML file (<ProjectName>.movrealtimedb) you will find the attributes which consent to modifying the characters used in the OPC Server's 'Tag Explorer' window for dividing the various components in connection strings which return after selecting an OPC item. These attributes are not visible in the Properties Window in the development environment, but can be modified by opening the <ProjectName>.movrealtimedb file:

```
...
<EnableOPCServer OPCServerName="" OPCServerDescription="MOVICON OPC Server"
EnableOPCServerAE="0" EnableOPCServerDynTag="1" OPCServerDebugEvents="0"
OPCServerAutoShutdown="0" OPCServerAutoUnregister="0" OPCServerAutoRegister="0"
OPCServerRefreshRate="250" OPCServerShutdownClientsTimeout="5000"
OPCServerEnableAEAck="0" OPCServerThreadingMode="1"
SINGLE_PATH_SEPARATOR="/" DOUBLE_PATH_SEPARATOR="\"
OPX_PATH_SEPARATOR="~">0</EnableOPCServer>
...
```

The properties involved are:

- **SINGLE\_PATH\_SEPARATOR:** character used for distinguishing the name of the OPC Server from the other items (ie. "[OPC]MMIOPC.Simulator\Simulator.CosDouble"). The default value is "/"
- **DOUBLE\_PATH\_SEPARATOR:** character used for distinguishing the name of the computer in which the OPC Server is located (ie. "[OPC]\\PCTest\\MMIOPC.Simulator\Simulator.CosDouble"). The default value is "\"
- **OPX\_PATH\_SEPARATOR:** character used for distinguishing the name of the OPC XML Server from the other items. The default value is "~"



It might be necessary to modify the values in these attributes when the OPC Server to which is being interfaced returns the same characters.

## Entering New Groups

---

The next step to take after having inserted an OPC Server is to establish a connection by entering one or more Groups which refer to that specific Server.

This operation can be done by selecting the "Add New OPC Group" item from the "Project Explorer's" command window or from the menu which appears after having double-clicked the right mouse key on the name of the interested OPC Server. The possibility of inserting more than one group will optimize communication further: each group has its own parameters which can be setup and can be enabled or disabled whenever needed by using the basic script functions.

To change the OPC Group properties, select the object with the mouse and use the Movicon **"Properties Window"**.

### Name

This edit box allows you to set the name to be assigned to the Group.

### Update Rate

This edit box allows you to set the update time in milliseconds of the Items with in the Group. This setting can be changed in Runtime with the appropriate basic script functions.



This update time is a parameter which is passed to the Server. The Server can manage it as much as is possible. If for example, the update time is too short the Server will use its own update rate.

### Update Rate NOT In Use

This edit box allows you to set the refresh time in milliseconds of the Items within the Group when the variables associated to the Item are not in use. This setting can be changed in Runtime with the appropriate basic script functions.



This refresh time must be set longer than the "Refresh Every" time.

### Deactivate NOT In Use

This option box allows you to deactivate the Group when the variables associated to the Items are not in use. This setting can be changed in Runtime with the appropriate basic script functions.

### Active

This option box allows you to enable or disable the Groups and as a consequence the Items it contains as well. This setting can be changed in Runtime with the appropriate basic script functions.

### Dead Band

This edit box allows you to set a Dead Band rate value from 0 to 100 for the Items contained in the Group. The dead band is applied to the Group's Items which have a EU Type parameter set on Analog, in this case the Low and High EU parameters are used for calculating the Item's range. The range is multiplied by the dead band rate to create an exception limit in cases in which the difference between the previous value read and the new one is higher.

The dead band is used for avoiding problems created due to interference in reading analogic values: with exceptions, the Item remains at the previous value read.

### Time Bias

This editbox allows you to set a value in minutes which will be used to convert the Item's Time Stamp property, contained within the Group, to the device's local time. Normally, there is no need to change the default value which is set by Movicon based on the PC's local time zone. For instance, using Rome's local time, being GMT + 1.00 hr, the Time Bias will be set at -60, an hour less than GMT.

### Local ID

This edit box allows you to set the Group with a numeric ID value to localize the strings sent by the Server. Normally the default zero value does not need changing.

## Entering New Items

---

One or more Items can be inserted into each Group and linked to project variables. The Item from the Server of interest is selected through the "Tag Browser" window which is displayed by selecting the "Add New OPC Item" from the Project Explorer's Commands window or from the menu which displays when double-clicking with the right mouse key on the interested Server. One or more tags can be selected at the same time by multiple selecting them from the "Tag Explorer" window and when confirmed with "OK" all the items will be inserted and Movicon will automatically create the variables and link them to the items. The variables will be created of the same OPC Server tag type and with the name defined as:

**<Name OPC Server>\_<Name OPC Items>**

All the characters not included in the name of a variable and which are found in the name of the OPC Server or in the name of the OPC item will be replaced with the underscore ("\_") character.

When customizing the name of the Movicon variable to be linked to the item, you will need to change the names of those inserted automatically or create them again and link them by means of using the item's appropriate properties.

To change the OPC Item properties, select the object with the mouse and use the Movicon **"Properties Window"**.

### Item ID

This box is used for identifying the Item which you wish to add the group, this box is automatically filled in when the Item is selected from the list situated in the "Tag Browser" window.

### Access Path

It may be useful to enter the path for localizing the Server if requested by the OPC Server.

### Variable

This list box allows you to select a variable from the project RealTime DB to be associated to the Server's OPC Item. While the item is being inserted Movicon will insert the variable in automatic mode by creating it with the <Name OPC Server>\_<Name OPC Items> name.

### Type

This option lets you enter the variable type to be read from the Server. Normally it is advised to leave the original format. The possible selects are:

- **Default Server:** No conversions will be done following the read or write of an item when this option box is enabled. In this case we advise you to use a Movicon variable of the same type set in the Item properties
- **Short, Long, Float, Double, etc.:** According to the option box checked, the corresponding conversion will be carried out on the item, and therefore the project variable assigned, to contain the data in the format indicated here

### Write Item to Server

This selection, when set enables the item in write, therefore the variable changes in the Movicon project will also involve the writing of the item.

### ReRead Item

This option allows you decide whether to execute a sync read of the value each time a write is executed (synchronous or asynchronous). This may be needed when using OPC servers (such as Rockwell's RsLinx) which do not manage asynchronous notifications in the correct way.

### Write Sync

This option allows you decide whether to execute a sync (default) or async write. The async writes can be executed very fast, in respect to those in sync, but the OPC Server must be setup to able to accept a very high number of requests, for example, when the variables in the Movicon project change value continuously. It is for this reason that the

default value has been set at Synchronous which is slower and more reliable and works well with all OPC Servers.

#### Read Item Startup

This option allows you to decide whether to execute an item read at the project startup, after it has been created, to synchronize the value of the variable with that from the field. This functionality is disabled for default to avoid any delays in initializing the OPC communication due to the fact that all items have been set at synchro. You can enable those properties in the items where it is really necessary to synchronize values.

## OPC Server

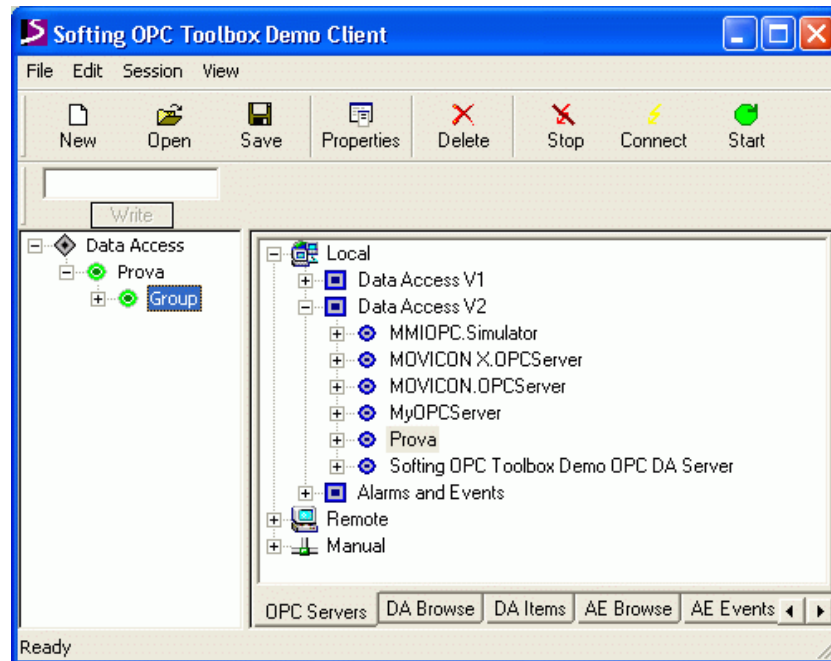
---

In addition to the OPC Client Movicon has also integrated the OPC Server functions which gives notifications of events to any OPC Client applications connected. The Movicon OPC Server supports Clients created with the "DataAccess" and "Alarms&Events" OPC standards.

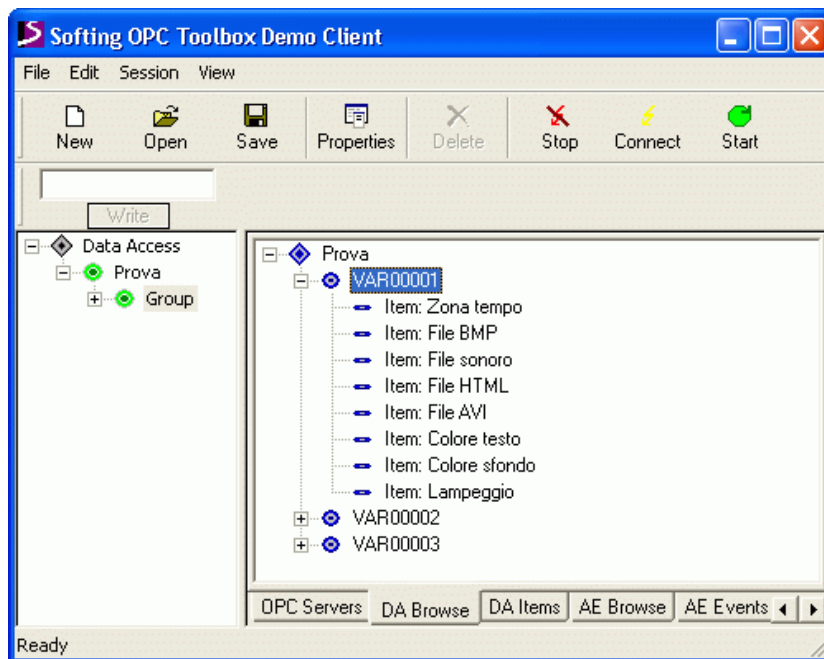
This technology permits the sharing of project variables with other applications with a OPC Client DA and at the same time it can notify a Client AE of events such as Alarms and Messages system information and variable changes in the applied project real-time Database.

Any OPC Client wishing to connect to the Movicon Server, according to the settings given in the project, is provided with a list of items to which it can connect to. The list is contained in a group identified by the project name and each item has the same name of the project variable to which it is linked. The possibility to change values is controlled within the project in the settings of each variable permitting the Movicon variables to be viewed externally with the maximum security.

Based on the settings implemented in the project, the Movicon OPC Server will startup automatically the moment a Client connects and likewise it will shutdown automatically when the Client disconnects.



When selecting the Movicon Server you will be provided with a list of the variables enabled for being shared with Clients.



## OPC Server Settings

The Movicon OPC Server can be configured through the **"Real Time DB OPC Server Settings"** which is accessed from the Real Time DB resource properties.

The Movicon OPC Server, if registered on the list of OPC Servers available, can startup in automatic, with the last opened project, the moment an OPC Client connects. To enable this function you need to disable the **"AutoUnregister Server"** function.

When selecting the **"AutoShutdown"** option, Movicon will shutdown at the same moment in which the OPC Client disconnects.

## OPC Server Variable Properties

To connect the variables through the OPC Server to OPC Client applications you need to setup the variables to be connected by means of using the **"Variable Options Properties"** after having configured the Movicon OPC Server settings. By doing it in this way the variables will be ready for Clients to create communication Tags.



The Tag is only a link between the Movicon variable and the OPC Client's item and nothing else. In order to obtain this link you must first enable the "Enable OPC Server" property of the Movicon variable so that the OPC Client can see the list of the Server variables.

# 31. On-line Changes to Projects

---

The Movicon developer may need to make changes to his or her project without interrupting it during Runtime mode as they would normally have to do. Movicon permits On-line changes to various resources of the project, such as Screens, Menus, Accelerators, Basic Scripts, etc.

To make on-line changes during the project Runtime mode you will need to open another Movicon instance, or rather start the Movicon application again. From the second active Movicon instance, select the Open command from the File menu. Select the project you wish to edit, which should be the one already running, from the window which appears. Once you have made the changes you will need to save the project. This should of course be consented by the protection key (License) being used. When the application in Runtime processes the edited resource, it will be loaded by the project and opened with its new changes. This means that the resources which were active when the changes were made will not acknowledge these changes if not unloaded and then reloaded in Runtime.



The project in which will put the changed into effect On-Line must be started up and executed directly in Runtime with a shortcut to Movicon .exe and option/R, with a shortcut to MoviconRunTime .exe or as service. Otherwise, if the project is run from a Movicon Development request any changed made On-Line to the resources will be lost.

Project On-line changes can be also done between two PCs connected up in ethernet network. In this case the Movicon development session should open the project residing on the RunTime station by means of using the ethernet network.



**On-Line Changes are not supported in WinCE. There is no way of knowing which file has changed with WinCE, therefore it will be necessary to reload all the project's files.**

## List of editable resources in RunTime

---

Projects can be edited with Movicon while being run. The changes made on-line are acquired straight away during and throughout the project run. The resources which can be edited on-line have been listed below. Those which cannot be edited on-line are not listed.

### IL Logic

Project IL Logic on-line editing is supported for screens and objects.

When any editing is done to the project's IL Logic the Runtime session will temporarily stop for the time necessary to load the changes.

When changes are made to the screen's IL Logic or one of its objects, you will need to reload the screen in memory again to acquire the changes made.



The window for the project's IL Logic debugging does not fully support the changes which can however be monitored with the local IL Logic debug.

### Real Time DB

Variables and Structure prototypes can be added.

The eliminated variables continue to be managed by the Runtime session until the next application startup.

Not all of the variable properties support on-line changes. These properties are:

- "General -> Type" Property
- "General -> Area" Property
- "General -> Address" Property
- "Options-> Enable OPC Server" Property

- "Options-> Enable Network Server" Property

**Menu Resources**

The on-line changes are applied to these resources after the first consecutive change page has been executed.

**Screen Resources**

The on-line changes are applied when the screen is reloaded into memory. If the screen was already active when the changes took place, you will have to close it, wait until it is unloaded from memory and then re-open it.

**Shortcut Resources**

The on-line changes are applied to these resources after the first consecutive change page has been executed.

**Basic Script Resources**

The Basic Script must be unloaded from memory with the "Stop" command in order to acquire the changes at the next startup.

**String Table**

The changes made in development mode for each column are loaded only when the column is activated in the Project in Runtime mode.

## 32. Movicon Configuration Keys

---

*The Movicon configuration keys permit you to customize some of the settings not available in the programming environment.*

There are a few advanced options within the Movicon environment which can only be modified when setting certain values using the keys in the Windows configuration registry or when editing ".ini" files with XML structures. You must take into account that in cases where the ".ini" file has been inserted, Movicon will load the configuration values from the file and not from the Windows configuration registry at startup. Movicon will only read values from the Windows configuration registry when no ".ini" file has been created.



**The Windows configuration registry is an important component for running the operating system and any errors could effect how it runs. Therefore, it is advised that you always use the XML file for modifying the configuration keys.**

### Editing XML Configuration files

---

The XML structure of the configuration file allows you to insert values in various keys as well. This structure provides a XML root tag called "Settings" in which all the Movicon sub keys with their respective values must be inserted. The file structure will result as follows:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<Settings>
<General>
<NumericPadFont>MS Sans Serif</NumericPadFont>
<MaxLogFiles>5</MaxLogFiles>
</General>
<OPCClient>
<WriteMaxBuffer>1</WriteMaxBuffer>
</OPCClient>
</Settings>
```

In this example two values have been inserted in the "General" key and one value has been inserted in the "OPCClient" key.

The XML files, according to the Movicon module used, must have the following names:

- **Movicon.ini** (se si utilizza il modulo Movicon.exe)
- **MoviconRunTime.ini** (se si utilizza il modulo MoviconRunTime.exe)
- **MoviconService.ini** (se si utilizza il modulo MoviconService.exe)

The XML files listed above must be inserted in the Movicon installation file, being where the executable files reside.

All the key values which can be inserted in the configuration files are listed and described in the appropriate tables found in the "**Configuration Key List**" section.

### Registry Key Editor Window

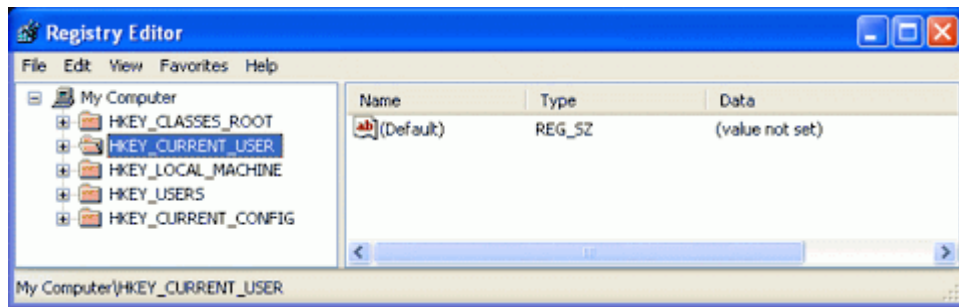
---

Windows configuration Registry key editing must be carried out with the Movicon program closed by carrying out the below specifications.

Go to the Windows application bar by means of the **Start** menu, or activate the **Run** command whereby a dialog window will display. Enter the '**Regedit**' command by using the keyboard and press **Enter**. This will open the System Registry Editor window which is

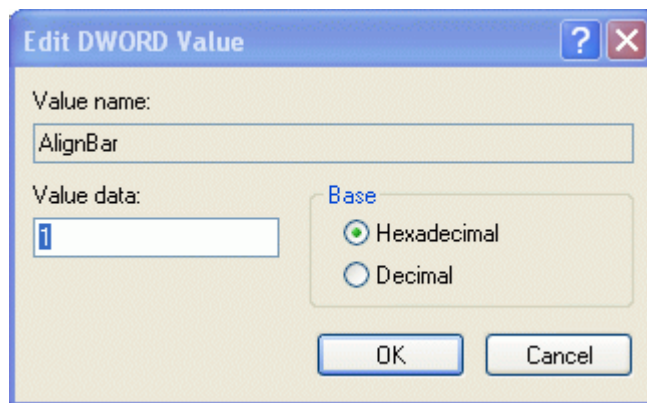


divided into two panes: the main items are listed in a tree structure of key classifications on the left, the values of the key selected on the left are shown in the right pane.



When referring to the table describing the keys mentioned in the "**Configuration Key List**" paragraph, you should select the key you are interested in by browsing through the tree structure on the left in the window. Once you have found the key, select it to view its already set values in the right pane. If the value you are interested in has not been set you can add it by activating the **New** command found in the **Edit** menu or with a right mouse click. The value you wish to insert must be one of the types specified in the key description table ("Type" column).

Any changes to just inserted or already existing values can be made by selecting the **Edit** command from the **Edit** menu or by double-clicking on the name of the value. In the window which appears you select the **Hexadecimal** or **Decimal** data Base and enter the value.



All the key values that Movicon uses can be searched for or inserted starting with this path:

**HKEY\_CURRENT\_USER\Software\Progea Automation 11.0\MOVICON 11[version]\**

# Configuration Key List

The registry keys listed below refer exclusively to the running of the Movicon program, and are therefore not influential to the running of the system.  
All the keys listed and described below can be searched and inserted starting from the following path:



Please bare in mind that the following keys listed below can be used with the last Movicon version but some of them may not be supported by previous versions.

**You must also take into consideration that the major part of the keys listed below are not presented in the configuration file or Windows registry. In fact Movicon does not create them in the installation phase. When a key is not present Movicon uses the default value, which is shown in the Default column in the tables listed below. Therefore it is at the programmer's discretion to create the keys and associate them to the values desired if different from the ones for default.**



The Windows registry keys are not used in the Movicon CE version to read the attributes which allow modifications to change the way they work. In order to modify the configuration key default values you must use the ".ini. Configuration File" as described in the relating Movicon section for WinCE.

**Registry Keys Development**  
**Registry Keys User Interface**  
**Registry Keys Communications**  
**Registry Keys Historicals**  
**Registry Keys Generals**  
**Registry Advanced Keys**  
**Speech Synthesis Keys**  
**SourceSafe Keys**  
**MS ActiveSync Keys**

## Registry Keys Development Registry Keys

List of parameter which influence the development environment.

Key	Value	Type	Description	Default	Win 32	Win CE
General	AppLook	DWORD	Application working environment look:  0 = VS.NET 1 = Office 2003 2 = Office XP	2	●	-
General	MaxUndoLevel	DWORD	Number of undo or redo levels of operations executed in project design mode.	20	●	-
General	CreateSharedVariables	DWORD	Area type where to insert new variables:	0	●	-

			0 = Not Shared 1 = Area Flag			
General	NotSharedRetentive	DWORD	"Retentive" option value when a new variable is inserted:  0 = Not Retentive 1 = Retentive	0	●	-
General	MRUCount	DWORD	Maximum number of projects listed in recent files under the "File" menu item and in the "Recent" tab after executing the "File-Open" command.	8	●	-
General	Trace	DWORD	Shows the identifiers called by the application in the Output window's "Trace Help" folder of the argument selected with the Dynamic Help window when active.  0 = Enabled 1 = Disabled	1	●	-
General	DefaultAlphaBlend	DWORD	Transparency percentage factor of the application windows. Valid only with Windows 2000 OS or later.	80	●	-
General	AnimationDelay	DWORD	Trigger in milliseconds for showing script editor windows.	1	●	-
General	AnimationTime	DWORD	Time in milliseconds for showing controls. Not used in application resources for the time being.	100	●	-
General	ContInser	DWORD	Enables the continuous insertion of objects on screens.  0 = New selection must be made from the 'Object' window after each object is inserted. 1 = Enabled	0	●	-

General	NumWatchWindows	DWORD	Number of watch Tabs in the Project Debug window (Watch window) when the project is run from the development mode (minimum value = 1).	4	●	-
General	RulerSize	DWORD	Screen side ruler widths in pixels.	18	●	-
DoNotAskAgain	*.*	DWORD	Values that deactivate confirm request windows, which appear after certain operations are carried out in programming, are saved in this key. If the "Do not ask again" option is marked before these windows are closed, the window in question will no longer be shown again.  To restore the initial situation just cancel the values in this registry key.		●	-
General	StringSep	SZ	/ character used for separating string table fields, when imported or exported from text files.	TAB	●	-
General	StoreCryptProject	DWORD	Sets the predefined value which the "Crypted Project" property must obtain a new project is being created.  0 = New projects will not be saved crypted 1 = New projects will be saved crypted	0	●	-
General	StoreCryptProjectResources	DWORD	Sets the predefined value which the "Crypted Project Resources" property must obtain when a new project is being created.  0 = New projects will be saved without crypted resources.	0	●	-

			1 = New projects will be saved with crypted resources.			
General	StoreUnicodeProject	DWORD	<p>Sets the predefined value to the "Unicode Project" property, when a new project is created.</p> <p>0 = New projects will not be saved in unicode. 1 = New projects will be saved in unicode.</p>	Standard language operating system: 0 Unicode language operating system: 1	●	-
General	StoreZippedProject	DWORD	<p>Sets predefined value to the "Zipped Project" property when creating new project is created.</p> <p>0 = New projects will not be saved compressed. 1 = New projects will be saved compressed.</p>	0	●	-
ExportCE	CheckFileTimeAndDate	DWORD	<p>Enables data control of files while exporting projects in WinCE.</p> <p>0 = Disabled 1 = Enabled</p>	0	●	-
General	DefaultNewNumInput	DWORD	Sets the predefined byte number value for the Input area (shared area), when creating a new project.	32768	●	-

			Note: minimum value = 10.			
General	DefaultNewNumOutput	DWORD	Sets the predefined byte number value for the Output area (shared area), when creating a new project.  Note: minimum value = 10.	32 76 8	●	-
General	DefaultNewNumFlag	DWORD	Sets the predefined byte number value for the Flag area (shared area), when creating a new project.  Note: minimum value = 10.	32 76 8	●	-
General	MaxStatObjects	DWORD	Max. number of Handle objects used for project statistics when running projects from the developing environment.	10 00	●	-
General	MaxArraySize	DWORD	Max. size to be obtained by variables in byte arrays with fixed lengths.	40 96	●	-
General	DisableDockedAnimation	DWORD	Disables the docked window animations. Therefore, when the mouse is positioned a tab, the corresponding window will appear on screen already open.  0 = Docked window animation is disabled. 1 = Docked window animation is enabled.  Note: Animations of these windows remain disabled when working in a server terminal session or when the symbol library window is open.	0	●	-

# User Interface Registry Keys User Interface

List of parameters that influence the user interface and graphics.

keys	Value	Type	Description	Default	Win 32	Win CE
General	TooltipTimeAutoPop	DWORD	Time in milliseconds for closing a tooltip window automatically.	5000	●	-
General	TooltipTimeInitial	DWORD	Time needed in milliseconds before a tooltip window can reappear in an object. the mouse pointer must remain outside the object for the time indicated in this parameter.	200	●	-
General	TooltipTimeFadeIn	DWORD	Fading in duration index time effect of the tooltip window while appearing in objects.	1	●	-
General	TooltipTimeFadeOut	DWORD	Fading out duration index time of tooltip window while disappearing from objects.	1	●	-
General	BlinkLedKeyboard	DWORD	Permits you to choose whether or not to have the "Scroll Lock" led blink when an alarm occurs.  0 = Blink disabled 1 = Blink enabled Note: Activating the blink might compromise application stability in devices which do not use standard drivers for managing keyboards.	0	●	●
General	BlinkTypeFast	DWORD	'Fast' option blink time in milliseconds, selectable in some symbol animation properties.	500	●	●
General	BlinkTypeMedium	DWORD	Blink time in milliseconds for the 'Medium Blink' option, selectable in some symbol animation properties.	1000	●	●

General	BlinkTypeSlow	DWORD	Blink time in milliseconds for the "Slow Blink" option, selectable in some symbol animation properties.	2000	•	•
General	NumberShade	DWORD	Number of colour shadings when using backgrounds with two colours fading into each other.	Win32 : 64 WinCE : 16	•	•
General	BrightnessOffset	DWORD	Number of brightness tones when using backgrounds with one colour fading.	50	•	•
General	StepBrightness	DWORD	Number of Brightness steps used when increasing or decreasing design brightness or contrast.	5	•	•
General	TraceCommentFont	SZ	Font used in standard variable "Trace Comment" windows.	MS Sans Serif	•	•
General	TraceCommentFontSize	DWORD	Font size used in variable "Trace Comment" windows.  Note: The size of the window on screen is in relation to this value. The size "8" is the minimum size for the default font. For reduce the window dimension it's necessary to use a different font, such as the "Small Font".	8	•	•
General	DBFilterFont	SZ	Font used in "Filter" window which can be opened from the "Historical Log", "DataLogger/Recipe" and "DB Trace" windows.	MS Sans Serif	•	•
General	DBFilterFontSize	DWORD	Font size in the "Filter" windows which can be opened from the "Historical Log", "DataLogger/Recipe" and "DB Trace" windows.  Note: The size of the window on screen is in relation to this value. The size	8	•	•



			"8" is the minimum size for the default font. For reduce the window dimension it's necessary to use a different font, such as the "Small Font".			
General	NumericPadFont	SZ	Font used for the numeric pad window.	MS Sans Serif	•	•
General	NumericPadFontSize	DWORD	Font size for numeric pad window.	Win32 /CE: 14  Pocket PC: 8	•	•
General	AlphaNumericPadFont	SZ	Font used for alphanumeric pad window.	MS Sans Serif	•	•
General	AlphaNumericPadFontSize	DWORD	Font size used for alphanumeric pad window.	Win32 /CE: 14  Pocket PC: 8	•	•
General	GetPasswordFont	SZ	<p>Font used for the password window for authenticating users.</p> <p>NOTE:</p> <ul style="list-style-type: none"> <li>the chosen font type must support the sizes set in the "GetPasswordFontSize" value</li> <li>the image on the top left of the password window will not appear if this key is used. This will avoid any overlapping of edit boxes and texts of in the window when changing size</li> </ul>	MS Sans Serif	•	•

General	GetPasswordFontSize	DWORD	<p>Font size used for the password window for authenticating users. The size "8" is the minimum size for the default font. For reduce the window dimension it's necessary to use a different font, such as the "Small Font".</p> <p>NOTE:</p> <ul style="list-style-type: none"> <li>the image on the top left of the password window will not appear if this key is used. This will avoid any overlapping of edit boxes and texts of in the window when changing size</li> </ul>	Win32 /CE: 14  Pocket PC: 8	•	•
General	PasswordExpiredFont	SZ	<p>Font used for the expired password window for authenticating users.</p> <p>NOTE:</p> <ul style="list-style-type: none"> <li>the chosen font type must support the sizes set in the "PasswordExpired FontSize" value</li> <li>the image on the top left of the password window will not appear if this key is used. This will avoid any overlapping of edit boxes and texts of in the window when changing size</li> </ul>	MS Sans Serif	•	•
General	PasswordExpiredFontSize	DWORD	<p>Font size used for the expired password edit window for authenticating users. The size "8" is the minimum size for the default font. For reduce the window dimension it's necessary to use a different font, such as the "Small Font".</p> <p>NOTE:</p> <ul style="list-style-type: none"> <li>the image on the top left of the password window will not appear if this key is used. This will avoid any</li> </ul>	Win32 /CE: 14  Pocket PC: 8	•	•

			overlapping of edit boxes and texts of in the window when changing size			
General	ShowSIP	DWORD	Enables the appearance of the WinCE virtual keyboard when needing to edit values.	1	-	●
MouseCursor	Visible	DWORD	Enables the mouse cursor visibility when application is being run.	1	-	●
General	DontUseDecorativeFont	DWORD	<p>Sets the application to manage fonts in order to speed up graphics updating to the max. Fonts will appear less attractive graphically as a consequence.</p> <p>0 = Decorative fonts 1 = No Decorative fonts</p> <p>Note: When this parameter is set at 1, the following attributes are forced when a new font is created:</p> <p>IfOutPrecision=OUT_RASTER_PRECIS IfClipPrecision=CLIP_STROKE_PRECIS IfQuality=DRAFT_QUALITY</p>	1	-	●

## Registry Key Communications

List of parameters which influence communications: networking, drivers, OPC

Key	Value	Type	Description	Default	Win 32	Win CE
OPCBrowse	UseRegistry	DWORD	<p>Enables use of Windows configuration registry to search for installed OPC Servers. This shows listed those OPC Server 1.0 which still used this method.</p> <p>0 = Uses OPCEnum to make a list of the installed OPC Servers.</p>	0	●	-

			1 = Uses the registry to make a list of the installed OPC Servers.			
General	DeadSocketTimeout	DWORD	Timeout in minutes before closing inactive network client connections.	5	●	●
General	MaxTimeStartupDriver	DWORD	Startup time in milliseconds for loading communication driver. When time expires the project will startup even if the communication driver has not been loaded.	5000	●	●
General	DontRemoveOPCItems	DWORD	Enables the Don't Remove of OPC server items, when the variable is no longer in use in the project. This only happens when the variable has been configured with a OPC dynamic address. In this case the OPC Server item is created when the variable goes into use, and can be destroyed when the variable goes out of use, according to the value set in this parameter.	1	●	●
OPCClient	WriteMaxBuffer	DWORD	Buffer size of variables linked to opc server items. Each variable change linked to a opc server item gets buffered. Changes are then sent to the opc server one at a time until the buffer is emptied.  1 = buffer management is disabled.	1	●	●

# Registry Keys Historicals

---

List of parameters which influence project historicals.

Key	Value	Type	Description	Default	Win 32	Win CE
General	UseDefaultSQLServer	DWORD	<p>Preferential ODBC Driver for ODBC DSN project connections. the ODBC DSN connections are created both in development mode, with the appropriate commands, or at project startup, if not already present. The application will attempt to create the ODBC connections by using the preferential driver first then the "MS Access" driver if this fails.</p> <p>0 = "MS Access" ODBC Driver 1 = Driver ODBC "MS SQL Server"</p>	1	●	-
General	SSInstance	SZ	Name of MS SQL Server instance which must be used for creating the ODBC DSN connections.	(local)	●	-
General	ODBCQueryTimeout	DWORD	Timeout in seconds for extracting data from one historical. Once exceeded the operation in progress will be annulled and notified with a message printed in the log.	15	●	●
General	ODBCLoginTimeout	DWORD	Timeout in seconds for authenticating database login. Once exceeded the operation in progress will be annulled and notified with a message printed in the log.	15	●	●
General	TimeToWaitAfterODBCError	DWORD	Waiting Time in seconds before retrying a command towards the database which	10	●	●

			generated error.			
General	MaxLogEntries	DWORD	Max. size in Kbytes of each LOG file created by the application's trace.	100	●	●
General	MaxLogFiles	DWORD	Max. number of LOG files used for each trace type. Once this value is reached data will be recycled to the oldest file.	10	●	●
General	IMDBMaxHMemory	DWORD	<p>Max. size in Mbytes of memory which can be allocated to shared areas to manage InMemoryDB (IMDB) historicals.</p> <p>Once this limit is reached no other value will be recorded until memory space has been freed. A "IMDB - Internal error: Out of shared memory" message will appear in the system log.</p> <p>Min. Value Min = 1048576 Max. Value = Memory available in PC/Device</p>	<p>Win32 : 16777216</p> <p>WinCE : 4194304</p>	●	●
General	IMDBMaxLMemory	DWORD	<p>Max. size in Mbytes of memory which can be allocated to private areas to manage InMemoryDB (IMDB) historicals.</p> <p>Once this limit is reached no other value will be recorded until memory space has been freed. A "IMDB - Internal error: Out of shared memory" message will appear in the system log.</p> <p>Min. Value = 1048576 Max. Vaue = Max. memory which can</p>	<p>Win32 : 67108864</p> <p>WinCE : 4194304</p>	●	●

			be allocated for each process for the operating system being used (WinCE 5.0 = 32 MB).			
OPCClient	Trace	DWORD	<p>Enabling of registration in the OPC Client log of information relating to events received for value change items and items written to OPC servers.</p> <p>0 = Only base information is traced. 1 = All information is traced.</p>	0	•	•
General	ADOCEProvider	SZ	Name of provider used for connecting to databases in WinCE.	String = Microsoft.SQ LSERVER.MOBILE.OLEDB.3.0	-	•
General	ADOCEDDataSourceExt	SZ	Extension used for creating databases in WinCE.	.sdf	-	•

# General Registry Keys Generals

List of general parameters which effect the application in runtime.

Key	Value	Type	Description	Default	Win 32	Win CE
General	MaxAppInstances	DWORD	Maximum number of application instances which can be executed at the same time.  0 = Unlimited.	0	●	●
General	ILSleep	DWORD	Time in milliseconds between the execution of one group of instructions and the next. The group of instructions' size is set with "MaxILInstruction" parameter.  Note: In cases where the number of instructions in an object's IL logic is lower than this value, the sleep time set will still be respected.	Win32: 10  WinCE : 50	●	●
General	MaxILInstruction	DWORD	The number of instructions processed before sleep time goes into action, equal to the value set in the "ILSleep" parameter.	100	●	●
General	ILLogicPriority	DWORD	Priority of the thread which processes the project's IL logic. This parameter has no influence on the priority with which the IL logic contained in the screen or object is processed. This value can be set from 0 to 255, where 0 is given to the project's IL logic with the highest priority over the rest.  Note: WinCE is a multithreading and deterministic operating system, therefore when too many high priority project IL logic will block all the other O.S. threads which have lower priorities.	Win32: 0  WinCE. 255	●	●



General	TimerEventFrequency	DWORD	<p>VBA design "OnTimer" event execution frequency. The value entered corresponds to the number of times which the procedure must be executed a second.</p> <p>Note: Values 1 to 10 are allowed.</p>	2	•	•
General	InstallComponents	DWORD	<p>Enables the installation and registration of the ADOCE and SQL Server CE optional components at the application startup.</p> <p>0 = the ADOCE and SQL Server CE components are installed and registered only when the project being run needs them. 1= the ADOCE and SQL Server CE components are installed and registered at the application startup with any project.</p>	0	-	•

# Advanced Registry Advanced Keys

Advanced parameter List. Avoid modifying these parameters unless absolutely necessary for they may effect system behaviour.



Caution: These parameters should never need modifying but if circumstances make this unavoidable only expert personnel, in strict agreement with the Progea Technical Support team, must carry out the necessary modifications only.

Key	Value	Type	Description	Default	Win 32	Win CE
General	UseLFH	DWORD	Enabling the use of one the Windows XP and Windows 3003 Server functionalities called LFH (Low Fragmentation Heap). This functionality allows memory to be used more efficiently.  0 = Disables the use of LFH 1 = Enables the use of LFH	1	●	-
General	MaxMemoryLoad	DWORD	Percentage of fixed memory in use in the operating system whose situation will become critical when exceeded. When this threshold is exceeded, the application will enter into "Low memory condition".  Note: Please refer to the manual for further information on how the resources are managed when entering the "low memory condition" status.	0	●	-
General	MaxDTMessagesPurge	DWORD	Number of events per tick emitted on the project status bar.	20	●	●
General	MaxProcessingLogLine	DWORD	Max. number of messages a second printed in the output window and the LOG files.	100	●	●
General	SleepCounter	DWORD	Application performance optimization value.	5	●	●
General	GeneralTimeout	DWORD	Timeout for certain types of operations. For example, influences the max. time of a Basic Script expression evaluation associated to a control.	3000	●	●

General	MaxSymbolCache	DWORD	Max. size of the symbol caches. Vectorial symbols which do not contain animations or VBA code, are managed as bitmap images. 0 = Disabled. 0 = Max. Num. of objects that can be put in the cache.	0	●	●
General	UseOffScreenMem	DWORD	Enables the use of the memory map for managing screens. The use of the memory map allows some operations on screen to be down faster, such as zooming, however this will require additional memory use for each screen managed.	Win32: 1 WinCE: 0	●	●
General	MaxCacheFont	DWORD	Font management cache size. Therefore fonts can be stocked away in the cache for reuse instead of being recreated right from the beginning whenever needed.  0 = Cache manager disabled  Note: The font cache keeps a map to speed things up in the most simplest way possible. However, some of the most common font styles are also managed in the cache such as bold, underline etc.	0	-	●
General	MaxAvailVirtual	DWORD	Quantity of memory in bytes still available, that becomes critical when lowers. when the memory still available for programs in the WinCE operating system , or the VM memory allocated from the process (WinCE 5.0 max. 32MB), is lower than this value; the application entered into the "low memory condition" status. Note: Please refer to the manual for further information on how resources are managed in a "low memory condition".	5000000	-	●

## Speech Synthesis Keys

In this table you will find a list of Windows configuration registry keys concerning the Microsoft speech synthesis (**TTS - Text-To-Speech**).

The values in these keys can be changed with the "MsMouth.exe" tool found on the installation CdRom.



### Caution:

The "MsMouth.exe" program utility always writes all modifications in the U.S. key independently from the operating system language in use. Therefore when using operating systems in Italian you will need to copy the "Mode" binary key contents from the "Local PC" to the "Local Computer" manually.

Key	Value	Type	Description	Default	Win 32	Win CE
HKEY_CURRENT_USER \Software\Voice \VoiceText \Local PC	Mode	BYNARY	Phoneme settings which the application used with the operating systems in USA version or other.	-	●	-
HKEY_CURRENT_USER \Software\Voice\ \VoiceText\ Local Computer	Mode	BYNARY	Phoneme settings which the application used with the operating systems in Italian.	-	●	-

## SourceSafe Keys

List of parameters which influence SourceSafe integration settings.

Key	Value	Type	Description	Default	Win 32	Win CE
VSSSettings	UserName	SZ	User name to be used for connecting to the SourceSafe database. If not specified the name of the user login the operating system will be used. The default value is an empty string.	""	●	-
VSSSettings	Password	SZ	Password to be used for connecting to the SourceSafe database. Must coincide with the one set for the SourceSafe User. The default is an empty string.	""	●	-

VSSSettings	DefaultPath	SZ	<p>Path and name of the SourceSafe Configuration File (usually called "srcsafe.ini").</p> <p>When left empty, the default database set in the SourceSafe is used. The default value is an empty string which corresponds to the last databased opened with the sourceSage 2005 administrator program.</p> <p>Note: This key needs to be set when using the Visual Studio SourceSafe 2005 version in Italian, as no database has been predefined for the SourceSafe. The Visual SourceSafe 2005 in English does not have this problem.</p>	""	●	-
VSSSettings	DefaultRoot	SZ	<p>Path to created the project in the SourceSage database. The default value is "\$/". The path to set in this key can be read using the SourceSafe and displaying the folder's properties wished to be used as default.</p> <p>Note: path to be set must always end with the "/" character.</p>	""	●	-
VSSSettings	SymbolGallery	SZ	<p>Name of the SourceSafe project for the symbol library. The default value corresponds to the name of the product followed by the "symbols" suffix. The same folder tree structure, with the various symbol categories filed in the installation folder in the computer, will then</p>	""	●	-

			be created in the SourceSafe project.			
VSSSettings	CheckOutComment	DWORD	Consents the SourceSafe dialog window to be displayed for entering a comment every time a "Check Out for Edit..." command is executed in a project resource.	0	●	
VSSSettings	CheckInComment	DWORD	Consents the SourceSafe dialog window to be displayed for entering a comment every time a "Check In" command is executed in a project resource.	1	●	
VSSSettings	Enabled	DWORD	Consents to enable the Visual SourceSafe managment. The default value is 1. In cases where the SourceSafe managment is disabled, all commands inherent to the SourceSafe will result as disabled. Therefore, when opening projects shared with the SourceSafe, problems may arise when editing their resources because these resources will lbe blocked in "read-only" mode by the SourceSafe. This will need you to delete the SourceSafe project and restore all the project files as "read-write".	1	●	

# MS ActiveSync Keys

This is a list of parameters which allow you to customize some of the aspects concerning the Project "Upload/Download" management when the MS ActiveSync plugin is used.

Chiave	Valore	Tipo	Descrizione	Default	Win 32	Win CE
Platform	DeviceInstallPath	SZ	Path used for creating the "InstallDir" value in the Windows CE configuration registry. The "InstallDir" value contained the path in which the runtime has been installed and is created from the product installation packet. Gets used on project Start/Stop command.	SOFTWARE\Apps\Progea Automation MovXCE	●	-
Platform	DeviceExecutable	SZ	Name of runtime application installed in the Windows CE device. Gets used on project Start/Stop command.	MovXCE.exe	●	-
Platform	LocalDriverDeviceSetup	SZ	Relating path where files are searched for communication driver setups.	MovXCESetup	●	-
Platform	LocalPocketPCSetupPath	SZ	Relating path where files for PocketPC application setups are searched for. This path is used when the project Startup command fails and consents to finding the installation cab file for Windows CE pocketPC platforms.	MovXCESetup\MovXce.PPC.ini	●	-
Platform	LocalHPCDeviceSetup	SZ	Relating path in which files for setting up applications for the SDK Standard platform are searched for. This is used when the project's Startup command fails and consents the installation cab file search for Windows CE SDK standard	MovXCESetup\MovXce.SSDK.ini	●	-

			platforms.			
Platform	DeviceBootFile	SZ	Name of the boot file. This is used after project uploading during the search for the boot file.	\\MovXCE.boot	●	-





# Index

## A

---

Accelerators	
Accelerator Activation.....	402
Accelerator Example.....	409
Accelerators Properties.....	406
Inserting Accelerators.....	402
Keyboard Commands.....	403
Accelerators Properties	
Accelerator Command Execution Properties.....	407
Accelerator Command General Properties.....	406
Accelerator General Properties.....	406
Accelerator Properties.....	406
ActiveX, OCX, OLE	
Active X Execution Properties.....	391
Active X properties.....	391
ActiveX Editing.....	389
ActiveX General Properties.....	392
ActiveX Objects.....	387
Editing ActiveX using VBA.....	389
Editing OLE objects.....	393
Inserting an ActiveX.....	388
Inserting OLE objects.....	392
OLE Objects.....	392
Alarm Window	
Alarm Help in the Alarm Window.....	270
Alarm Window.....	267
Alarm Window Execution Properties.....	276
Alarm Window Field Choice.....	269
Alarm Window Fields.....	267
Alarm Window Style Properties.....	270
Alarms	
Alarms Properties.....	458
Inserting Alarms.....	453
Alarms Properties	
Alarm Threshold Event Notification Properties.....	465
Alarm Threshold Execution Properties.....	462
Alarm Threshold General Properties.....	460
Alarms General Properties.....	458
Alarms Properties.....	458
Alarms Style Properties.....	463

## B

---

Bugbase.....	31
Buttons and Hot Regions	
Buttons and Hot Regions.....	240

Buttons and Hot Regions Execution Properties.....	242
Components deriving from Button Objects.....	241

## C

---

Charts	
Charts.....	311
Charts Settings.....	311
Charts Properties	
Chart Execution Properties.....	318
Chart General Properties.....	317
Chart Properties.....	317
Chart Style Properties.....	319
Child Projects	
Child Project Resource Access.....	590
Child Project Users Inclusion.....	592
Child Projects Path.....	590
Client Profile	
Client Profile.....	608
Client Profile General Properties.....	609
ComboBox	
Combo Box Objects.....	262
Combo Box Style Properties.....	263
Combo Box Variables Properties.....	263
Command List for the Objects	
Alarms Commands.....	449
Change Language.....	448
Help Commands.....	448
Menu Commands.....	446
Report Commands.....	444
Screen Commands.....	439
Script Commands.....	442
System Commands.....	447
Users Commands.....	443
Variables Commands.....	435
Commands Schedulers	
Commands Scheduler Properties.....	426
Commands Schedulers General Properties.....	426
Example of Scheduling Commands.....	429

## D

---

Data Loggers	
Data Logger Properties.....	526
Data Loggers and Recipe ODBC Link.....	517
DataBase.....	512
Database Link to Report Files.....	513
DataBase Structure.....	513
Displaying Database.....	514

Editor Data Loggers and Recipes .....	516
Importing Data Loggers and Recipes from other Recipes.....	517
Inserting Data Loggers and Recipes.....	516
Data Loggers Properties	
Data Logger Column General Properties .....	529
Data Logger Execution Properties.....	529
Data Logger Recording Time Properties.....	528
Data Logger Style Properties .....	526
Data Loggers and Recipes common Column General Properties .....	525
Data Loggers and Recipes common Database Settings Properties.....	524
Data Loggers and Recipes common Execution Properties .....	523
Data Loggers and Recipes common General Properties.....	523
DataLogger Window	
DataLogger Data Filter.....	324
DataLogger Window.....	320
DataLogger Window Execution Properties .....	323
DataLogger Window Field Choice.....	321
DataLogger Window Fields .....	320
DataLogger Window Style Properties.....	322
Displays	
Display Style Properties .....	258
Display Variables Properties .....	256
Displays objects .....	256
Dynamic Variables	
Drivers Dynamic Variables .....	166
Dynamic Variables .....	163
Network Dynamic Variables.....	164
OPC Dynamic Variables.....	165

## E

---

Embedded Screens	
Embedded Screen Style Properties .....	254
Embedded Screens .....	254
Event Object Properties	
Event Object General Properties.....	432
Event Object Properties .....	431

## G

---

Gauges	
Components deriving from the Gauge object .....	247
Gauge Circular Properties.....	251
Gauge objects .....	245
Gauge Scale Properties.....	252
Gauge Style Properties .....	250
Gauge Variable Properties.....	249
Gauge Warning Zone properties.....	253
Graphic Interface	
Distribute Object Space .....	213
Graphic Editing Operations.....	210
How to create Graphic Interfaces .....	206
Importing/Exporting Draws .....	216
Multiple Objects Selection .....	211
Object Alignment Functions.....	211
Overlap Order .....	210
Tabulation Order .....	210
Grid	
The Grid.....	330

The Grid Buttons .....	330
------------------------	-----

## H

---

Historical Log	
Database of Historical Log .....	473
Historics	
DBMS Settings for Recording data on DataBase .....	488
Hour Selector	
Hour Selector .....	336
Hour Selector Execution Properties.....	341
Hour Selector Style Properties.....	338

## I

---

IMDB	
IMDB.....	485
IMDB Settings for Recording data .....	486

## L

---

ListBox	
List Box Objects .....	261
List Box Properties.....	261
Log Window	
DataBase Fields .....	279
Log Window.....	277
Log Window Data Filter .....	283
Log Window Execution Properties.....	282
Log Window Field choice.....	278
Log Window Fields .....	277
Log Window Style Properties .....	280
Logic IL (Instructions List)	
AND Logic of Variables .....	555
Arithmetic Instructions.....	565
Bracket levels for the CLR.....	552
Combined Logic Result CLR.....	551
Compare Instructions.....	563
Constants.....	554
End program processing.....	554
Floating Point Variables .....	553
Inserting Comments .....	554
Inserting Instructions into IL Logic.....	549
Jump to Labels.....	552
Load of Variables and Constants .....	559
Logic in Bit Instructions.....	555
Logic Operations on Word Variables .....	559
Logic Programming.....	551
Mathematic Instructions .....	568
Operations on Accumulators.....	560
Sequence Combination.....	555
Setting Timers.....	556
Setting Variables.....	555
Summary Table of Arithmetic Instructions .....	565
Summary Table of Bit Instructions .....	557
Summary Table of Comparison Instructions .....	563
Summary Table of Mathematic Instructions .....	568
Summary Table of Word Instructions.....	560
The ACCU1 and ACCU2 Logic Accumulators .....	559

<i>The Logic Accumulators .....</i>	<i>553</i>
<i>Transfer of Variables .....</i>	<i>559</i>
<i>Variable OR Logic .....</i>	<i>555</i>
<i>Word Logic Instructions .....</i>	<i>559</i>

## M

---

Menu	
<i>Customizing Menu Bars .....</i>	<i>398</i>
<i>Inserting a Menu .....</i>	<i>396</i>
<i>Menu Example.....</i>	<i>400</i>
<i>Menu Items .....</i>	<i>396</i>
<i>Menu Properties .....</i>	<i>398</i>
<i>Test Menu .....</i>	<i>398</i>
Menu Properties	
<i>Menu General Properties.....</i>	<i>398</i>
<i>Menu Item Execution Properties.....</i>	<i>399</i>
<i>Menu Item General Properties .....</i>	<i>399</i>
<i>Menu Properties .....</i>	<i>398</i>
Menu's Project	
<i>Context Resource Menus.....</i>	<i>84</i>
<i>Menu .....</i>	<i>72</i>
<i>Menu Draws.....</i>	<i>78</i>
<i>Menu File.....</i>	<i>72</i>
<i>Menu Help (?) .....</i>	<i>83</i>
<i>Menu Insert.....</i>	<i>77</i>
<i>Menu Layout.....</i>	<i>77</i>
<i>Menu Modify .....</i>	<i>74</i>
<i>Menu Tools.....</i>	<i>81</i>
<i>Menu View.....</i>	<i>75</i>
<i>Menu Windows .....</i>	<i>82</i>

## N

---

Network Client Settings	
<i>Network Client General Properties .....</i>	<i>606</i>
<i>Network Client Properties.....</i>	<i>605</i>
<i>Network Client Settings .....</i>	<i>605</i>
<i>OPC Client XML DA Properties.....</i>	<i>606</i>
Network Server Properties	
<i>HTTP Transport Advanced Settings</i>	
<i>Properties.....</i>	<i>603</i>
<i>HTTP Transport Properties.....</i>	<i>602</i>
<i>HTTP Transport SOCKS Options Properties</i>	
.....	<i>604</i>
<i>Local Transport Properties.....</i>	<i>599</i>
<i>Log and Security Properties.....</i>	<i>598</i>
<i>Network Server Properties .....</i>	<i>597</i>
<i>Network Services Properties .....</i>	<i>597</i>
<i>TCP Transport Advanced Settings Properties</i>	
.....	<i>600</i>
<i>TCP Transport SOCKS Options Properties.....</i>	<i>601</i>
<i>UDP Transport Advanced Settings Properties</i>	
.....	<i>602</i>
<i>UDP Transport Properties.....</i>	<i>601</i>
Networking	
<i>Network Connection .....</i>	<i>595</i>
<i>Network installing.....</i>	<i>594</i>
<i>Project Remote Debug.....</i>	<i>596</i>
<i>Windows Terminal Server.....</i>	<i>595</i>
New Project Wizard	
<i>Alarms Settings.....</i>	<i>50</i>
<i>Communication Driver.....</i>	<i>47</i>
<i>Data Loggers &amp; Recipes Settings (ODBC) .....</i>	<i>49</i>
<i>DataBase Settings (ODBC) .....</i>	<i>47</i>
<i>New Project .....</i>	<i>44</i>
<i>New Project Wizard Customizing .....</i>	<i>54</i>

<i>Project Constraints .....</i>	<i>51</i>
<i>Project Name.....</i>	<i>44</i>
<i>Screens .....</i>	<i>47</i>
<i>Users.....</i>	<i>45</i>

## O

---

ODBC	
<i>Connection Pooling .....</i>	<i>480</i>
<i>Creating ODBC links automatically .....</i>	<i>480</i>
<i>Creating ODBC Links Manually.....</i>	<i>480</i>
<i>ODBC Drivers .....</i>	<i>477</i>
<i>ODBC Links .....</i>	<i>478</i>
<i>ODBC Standard.....</i>	<i>476</i>
<i>SQL Server with MSDE .....</i>	<i>482</i>
OPC	
<i>Entering New Groups .....</i>	<i>623</i>
<i>Entering New Items .....</i>	<i>624</i>
<i>Entering New OPC Servers.....</i>	<i>621</i>
<i>OPC Client DA Properties .....</i>	<i>620</i>
<i>OPC Client Editor.....</i>	<i>619</i>
<i>OPC Server .....</i>	<i>625</i>
<i>OPC Server DA Properties.....</i>	<i>621</i>

## P

---

Project Settings	
<i>Child Project Settings.....</i>	<i>131</i>
<i>Execution Project Settings .....</i>	<i>124</i>
<i>General Project Settings.....</i>	<i>121</i>
<i>Heap Manager Settings .....</i>	<i>126</i>
<i>Historical Log Project Settings .....</i>	<i>129</i>
<i>Project Path Settings .....</i>	<i>122</i>
<i>Project Platform Settings .....</i>	<i>122</i>
<i>Project Settings .....</i>	<i>120</i>
<i>Project Spooler Manager Settings .....</i>	<i>132</i>
Project Windows	
<i>Dynamic Help Window.....</i>	<i>108</i>
<i>Dynamic Properties Inspector Window .....</i>	<i>100</i>
<i>Logic Explorer Window .....</i>	<i>93</i>
<i>Objects Window .....</i>	<i>96</i>
<i>Output Window .....</i>	<i>107</i>
<i>Popup Window .....</i>	<i>111</i>
<i>Project Explorer Window.....</i>	<i>89</i>
<i>Properties Window.....</i>	<i>90</i>
<i>Symbols Libraries Window .....</i>	<i>97</i>
<i>Watch Window.....</i>	<i>109</i>
Public Symbols	
<i>Public Symbols.....</i>	<i>381</i>

## R

---

RAS	
<i>RAS Station General Properties.....</i>	<i>607</i>
<i>RAS Stations .....</i>	<i>607</i>
Real Time DB Settings	
<i>Real Time DB General Settings .....</i>	<i>156</i>
<i>Real Time DB OPC Server Security Settings</i>	
.....	<i>159</i>
<i>Real Time DB OPC Server Settings.....</i>	<i>158</i>
<i>Real Time DB Retentive Data Settings .....</i>	<i>157</i>
<i>Real Time DB Settings.....</i>	<i>156</i>
<i>Real Time DB Trace DB Settings .....</i>	<i>160</i>

Real Time DB Variables InUse Management Settings .....	157
Real Time in Shared ODBC Settings .....	162
<b>Recipes</b>	
Creating Recipes in the Database .....	520
Recipes on Database .....	518
Recipes Properties .....	531
SQL Commands for Recipes in the Database .....	519
The Query in Recipes .....	522
<b>Recipes Properties</b>	
Data Loggers and Recipes common Column General Properties .....	525
Data Loggers and Recipes common Database Settings Properties .....	524
Data Loggers and Recipes common Execution Properties .....	523
Data Loggers and Recipes common General Properties .....	523
Recipe column General Properties .....	534
Recipe Database Settings Properties .....	533
Recipe Execution Properties .....	531
<b>Registry of Windows</b>	
List of Registry Keys .....	631
MS ActiveSync Keys .....	652
Registry Advanced Keys .....	647
Registry Keys Communications .....	640
Registry Keys Development .....	631
Registry Keys Generals .....	645
Registry Keys Historicals .....	642
Registry Keys User Interface .....	636
SourceSafe Keys .....	649
Speech Synthesis Keys .....	649
<b>Reports</b>	
Crystal Report™ .....	545
How to create a Report .....	537
Report Designer .....	535

## S

<b>Scaling Variables</b>	
Non Linear Scaling .....	422
Scaling Variables Properties .....	420
Variable Scaling General Properties .....	420
<b>Screens</b>	
Inserting a Screen .....	186
Local Screen Variables .....	191
Numeric and Alphanumeric Screens .....	192
Opening Screen Modalities .....	187
Screen Properties .....	197
Screens with Parameters .....	189
Startup Screen .....	186
<b>Screens Properties</b>	
Associating Variable Script Events to Screens .....	201
Screen Background Properties .....	199
Screen Execution Properties .....	201
Screen General Properties .....	197
Screen Properties .....	197
Screen Style Properties .....	198
<b>SoftLogic</b>	
ProConOs OPC Server Configuration .....	584
Soft Logic General Properties .....	582
Variable Database Synchronization .....	583
<b>Strings Table</b>	
Inserting Strings .....	411
Multilingual Functions .....	413
<b>Structure of a Project</b>	
Customizing Menu and Toolbars .....	60

Docking Windows .....	56
Grid .....	59
Group of Resources .....	112
Menu .....	72
Project Files .....	114
Rulers .....	58
The Resources .....	112
ToolBars .....	62
Workspace .....	55
<b>Structure Variables</b>	
Default Structures in symbols .....	170
Pointing to Structure Variables .....	170
Structure Prototypes General Properties .....	169
Structure Variables .....	167
Variable Member Order .....	169
<b>Support</b>	
Tech Support .....	30
Web Support Center .....	30
<b>Symbols</b>	
Creating or Modifying Symbols .....	221
Creating Symbols .....	220
Event OnCustomizeSymbol() .....	224
Graphic Editing of Symbols .....	219
Images inside Symbols .....	224
Inserting Symbols in Screens .....	223
Inserting Symbols into the Library .....	222
Script Code Editing .....	223
Symbols as Templates .....	221
Variables inside Symbols .....	224
<b>Symbols, Draws and Controls Properties</b>	
Access Levels Properties common to Drawings and Controls .....	376
Associating Variable Script Events to Symbols .....	380
Back Color Properties common to Drawings and Controls .....	366
Colour Threshold Settings in Drawings and Controls .....	369
Composed Movement Properties common to Drawings and Controls .....	353
Dragging Properties common to Drawings and Controls .....	378
Drawings and Controls common General Properties .....	349
Drawings and Controls Common Properties .....	345
Edge - Text Color Properties common to Drawings and Controls .....	367
End X Point Property common to Drawings and Controls .....	360
End Y Point Properties common to Drawings and Controls .....	361
Filling Properties common to Drawings and Controls .....	367
Font Properties common to Drawings and Controls .....	377
Image Properties common to Drawings and Controls .....	371
Move X Properties common to Drawings and Controls .....	355
Move Y Properties common to Drawings and Controls .....	356
Points Properties common to Drawings and Controls .....	357
Position Properties common to Drawings and Controls .....	374
Rotation Properties common to Drawings and Controls .....	363
Scaling Properties common to Drawings and Controls .....	362
Script Debug Output Properties common to Drawings and Controls .....	378
Start X Point Properties common to Drawings and Controls .....	358
Start Y Point Properties common to Drawings and Controls .....	359

<i>Stroke Attributes Properties common to Drawings and Controls</i> .....	375
<i>Style Properties common to Drawings and Controls</i> .....	346
<i>Text Properties common to Drawings and Controls</i> .....	364
<i>Visible Properties common to Drawings and Controls</i> .....	352
Synapses	
<i>Connecting Synapses</i> .....	575
<i>Inserting Synapse</i> .....	574
<i>Synapse Execution</i> .....	578
<i>Synapse Logic Programming</i> .....	577
System Variables	
<i>System Variables</i> .....	145

## T

Tool Bars	
<i>Aligning Bar</i> .....	66
<i>Formatting Bar</i> .....	68
<i>Layer Bar</i> .....	67
<i>RunTime Bar</i> .....	70
<i>Status Bar</i> .....	69
<i>Symbol Bar</i> .....	67
<i>Tool Bar</i> .....	62
<i>Tool Bars</i> .....	62
TraceDB Window	
<i>TraceDB Data Filter</i> .....	329
<i>TraceDB Window</i> .....	325
<i>TraceDB Window Execution Properties</i> .....	328
<i>TraceDB Window Field Choice</i> .....	326
<i>TraceDB Window Fields</i> .....	325
<i>TraceDB Window Style Properties</i> .....	327
Trends	
<i>Change Language for Trends Pens</i> .....	293
<i>I Trends</i> .....	285
<i>Keyboard with Trends</i> .....	291
<i>Links to Data Logger for Trends</i> .....	290
<i>Runtime commands for Trends</i> .....	292
<i>Template as vectorial Trends</i> .....	310
<i>Trends advanced features using VBA interface</i> .....	291
<i>Trends data visualization</i> .....	291
<i>Trends Features</i> .....	289
Trends Properties	
<i>Trend Pens Properties</i> .....	298
<i>Trends Execution Properties</i> .....	306
<i>Trends Grid Properties</i> .....	305
<i>Trends Legenda area Properties</i> .....	302
<i>Trends Pens Area Properties</i> .....	302
<i>Trends Settings</i> .....	294

<i>Trends Style Properties</i> .....	294
<i>Trends Time area Properties</i> .....	303
<i>Trends Variables Properties</i> .....	308

## U

Updates .....	31
User and Password	
<i>NT/XP Users Group Properties</i> .....	496
<i>User Levels and Access Levels</i> .....	490
<i>Users and User Groups Child Project Option Properties</i> .....	497
<i>Users and User Groups General Properties</i> .....	495
<i>Users and User Groups Properties</i> .....	495
Users and User Groups Properties	
<i>General Properties common to Users and User Groups</i> .....	493
<i>User Data Properties</i> .....	500
<i>User Group General Properties</i> .....	498
<i>User Group Properties</i> .....	498
<i>Users and User Groups Properties</i> .....	495
<i>Users General Properties</i> .....	499
<i>Users Properties</i> .....	499

## V

Variables (Tags)	
<i>Retentive Variables</i> .....	152
<i>Structure Variables</i> .....	167
<i>Tracing Variables</i> .....	142
<i>Variable Areas</i> .....	152
<i>Variable List(Tags)</i> .....	141
<i>Variables Properties</i> .....	171
Variables (Tags) Properties	
<i>Associating Alarms to Variables</i> .....	183
<i>Associating Data Loggers and Recipes to Variables</i> .....	183
<i>Associating Events to Variables</i> .....	183
<i>Variable Access Level Properties</i> .....	177
<i>Variable Engineering Data Properties</i> .....	174
<i>Variable General Properties</i> .....	172
<i>Variable Network Client Properties</i> .....	181
<i>Variable Options Properties</i> .....	178
<i>Variable Properties</i> .....	171
<i>Variable Real Time ODBC Properties</i> .....	181
<i>Variable Trace Options Properties</i> .....	179



## This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.